

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete
Computer and Automation Institute, Hungarian Academy of Sciences
Исследовательский институт вычислительной техники и автоматизации
Венгерской Академии наук

СТАТЬИ КОНФЕРЕНЦИИ КНВВТ ПО АВТОМАТИЗАЦИИ ИНФОРМАЦИОННЫХ
ПРОЦЕССОВ НА ПЕРСОНАЛЬНЫХ ЭВМ
Будапешт, 5-9 мая 1986 г.

Том I.

KNVVT CONFERENCE ON AUTOMATION OF INFORMATION PROCESSING ON
PERSONAL COMPUTERS
Budapest, May 5-9, 1986

Vol I.

Будапешт, 1986
Budapest, 1986

Tanulmányok 193/1986
Studies 193/1986

A kiadásért felelős:

DR. KEVICZKY LÁSZLÓ

Редактор:

ИШТВАН РАТКО

Editor:

ISTVÁN RATKÓ

ISBN 963 311 222 2

ISSN 0324-2951

87/64. Alfaprint
F.v.: Barabás Gábor

С О Д Е Р Ж А Н И Е

Стр.

Алексич Д., Гал А., Керекфи П., Ковач К., Лендел Т., Ракоци Ф., Руда М.: Госпитальная информационная система в всевенгерском кардиологическом институте. Схемное изложение системы	7
Барнев П., До Вьет Нга: Система автоматизированного составления свободных текстов	15
Борковский А.Б.: Текстовая база данных	27
Виткова Г., Гусек Д.: Интеграция баз данных и баз знаний	39
Гейци И.И., Карпачев Г.И.: Использование сетей ЭВМ при проектировании автоматизированной системы управления городским районом	53
Инзелт П., Таполцаи Л.: Имитационная модель для проверки реализуемости плана поставок при управлении продуктопроводом для транспортировки нескольких нефтепродуктов	73
Кирилова Р.К., Каравасилева Ю.П.: Базовые элементы в автоматизации информационных процессов административно-управленческих систем	101
Ле Мань: Некоторые вопросы о сочетании математического обеспечения для локальной сети на базе микро-ЭВМ	111
Марков Кр.К.: Система автоматизированного обслуживания информационного взаимодействия	121
До Вьет Нга: Система автоматизированного составления свободных текстов	137

Сахно С.: Компиляция знаний в экспертной системе ЭСАК	145
Стогний А.А., Глазунов Н.М.: Проблемы интеграции в базах данных	155
Занг Конг Тхе: Руководитель в автоматизированном учреждении	167
Трифорова З.: Расширение системы LISPKIT: интерпретатор функциональных программ в логическом стиле посредством унификации	177
Хенчей Г.: Структуризация Булевых данных на персональных компьютерах	193
Цекова Т.: Реализация системы контроля за исполнительской дисциплиной на основе процедурной модели	201
Шкуртов В., Керпеджиев С.: Абонентная система для корреспонденции	215

ПРЕДИСЛОВИЕ

Конференция КНВВТ /Комиссия Научных Вопросов Вычислитель-
ной Техники/ по автоматизации информационных процессов на пер-
сональных ЭВМ состоялась в Будапеште с 5 по 9 мая 1986 г.

Тематики конференции были следующие:

- организация и проектирование данных,
- язык обучения,
- прикладное программное обеспечение,
- технологии прикладного программирования в автоматизированных информационных системах, системах обработки данных и системах принятия решений, экспертных системах и системах баз знаний.

Актуальными сферами применений названных выше исследований и разработок являются

- научные исследования,
- административная деятельность,
- обучение.

Этот сборник содержит в себе статьи представленные на кон-
ференцию. Число статей было так велико, что сборник надо было
разделить на два тома. Первый том содержит статьи на русском
языке, а второй - статьи на английском языке.

Иштван Ратко

ГОСПИТАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА В
ВСЕВЕНГЕРСКОМ КАРДИОЛОГИЧЕСКОМ ИНСТИТУТЕ

Схематическое изложение системы

Алексич Д., Гал А., Керекфи П., Ковач К.,
Лендел Т., Ракоци Ф., Руда М.

Конфигурация: локальная сетевая система, в узлах которой
расположены 8-разрядные микромашины.

Микромашина: SYSTER (сделано в ИИВТА ВАН)

Z80 CPU

128 KBYTE RAM

2 * 1 MBYTE FLOPPY

PANASONIC PRINTER

V24 INTERFACE

MOD81-M: микромашина, работает в лаборатории, она накапли-
вает данные измерения хематологического автомата
(сделано в МЕДИКОР).

Устройство сетевой связи: ЛАНБЛОКС (ИИВТА ВАН).

Скорость передачи данных в сети: 1 Мбит/сек.

Математическое обеспечение: (сделано в ИИВТА ВАН)

Операционная система: НЕТТИ

Сетевая система: КАПТАЛАН

Система управления базой данных: МИКРО-ШИВА

Пользовательские подсистемы:

Система приёма больных:

- приём данных, их передача на отделы госпиталя;
- изготовление регистров, списков;
- помощь в административной работе;

Система лабораторных работ:

- хранение и обработка данных измерений на ЭВМ;
- печат данных осмотра;
- помощь в административной работе;
- передача данных на отделы госпиталя с помощью сетевой системы;

Система на отделе госпиталя:

- приём данных больной на отделе;
- хранение описания состояния больной;
- помощь в приготовлении данных осмотра
(вычисление, печат)
- хранение диагнозов;
- разработка лабораторных данных;
- регистр медицинских средств;
- возможность обращения к данным архива;
- составление заключительного отчёта;

МИКРО-ШИВА - Система управления базой данных:

ШИВА - универсальная система, эффективная и для пользователей и для специалистов.

Пользователь без трудности переходит от обычной административной работы к работе на ЭВМ, поскольку ШИВА в максимальном мере освобождает его от манипуляции над данными. Это означает то, что на мониторе появляется незаполненный бланк и навигация в нём делается свободно, при вводе ошибочных данных моментально дает сигнал.

ШИВА - модулярная система, поэтому специалисты имеют большой степень свободы в разработке пользовательских систем.

Система ШИВА успешно применяли и на других областях:

- прочие применения на других областях здравоохранения (Фабрика медицинских средств);
- промышленное применение (автомашинный завод);
- применения в административной работе.

Сетевая система:

Передача данных в сетевой системе идёт автоматически, в этом не употребляется пользовательской помощи. Передача данных идёт между узлами распределённой базы данных под контролем правил, изложенных в описании распределённой базе данных. Благодаря этому, правила передачи данных независит от функции отдельных узлов.

В пользовательской системе, употребляемая системой ШИВА, осуществляется преимущество распределённой базы данных:

- удастся разработать мощную пользовательскую систему, употребляя ЭВМ с простоватой (дешовой, малоёмкостной) конфигурацией;
- в каждом узле достигаются данные, введённые в других узлах, в то же время каждый узел имеет независимость благодаря собственным составом данных.

СИСТЕМА ПРИЁМА БОЛЬНЫХ

ФУНКЦИИ СИСТЕМЫ

Приём данных: идёт в соответствии требованиям принятых на отделах бланков и принятых в стране бланка "история болезни".

Характеристики:

- моментальное наблюдение ошибок, предупреждение;
- аналитическая передача данных на отделы больницы (настоящее время ЭВМ находится лишь на отделе детских больных);
- составление потребных документов при приёме (история болезни, статистический листок);

Составление дневного составного донесения: для каждого отдела получается регистр дневного движения больных (приём, уход, смерть, перемещение). Регистр помогает администрации на отдела.

Составление месячного составного донесения: месячное накапливание дневных изменений в составном донесении с дополнительными показателями (численность лежащих стационарных больных, число смертных случаев, среднее время излечения, загрузка отдела).

Список больных для швейцарской: показывает моментальное размещение лежащих стационарных больных в алфавитном порядке. Содержит данные и тех больных, которые на 3 дня назад удалялись. Этот список находится на швейцарской после того, как закончивается приём больных.

Список лежащих стационарных больных: содержит данные больных каждого отдела в алфавитном порядке.

Архив данных: данные уходящих больных переписываются на архив-плёнку. Составляется и список о содержании плёнки и так **есть** возможность поискать больных по имени и по персональным регистрационным числу.

ЭКСПЛУАТАЦИЯ СИСТЕМЫ

Приём больных идет непрерывно - прежде всего в первой половине дня. Сетевой связь составляется в паузах приёма данных.

Предоставляющие программы эксплуатируются или раз посуточно (дневное составное донесение, список больных для швейцарской), или реже (месячное составное донесение, список лежащих стационарных больных, архив данных).

СИСТЕМА ЛАБОРАТОРНЫХ РАБОТ

ФУНКЦИИ ВНАЧАЛЕ РАБОТЫ:

- получение образцов, их ввод;
- внесение запросов лабораторных измерений в гросбух;
- сбор тех измерений, которые наиболее часто потребуется сделать (рабочий список);
- приготовление список для отделов лаборатории.

ФУНКЦИИ В ПЕРВОЙ ПОЛОВИНЕ ДНЯ:

- внесение непрерывно возникающихся запросов в гросбух;
- дополнительное приготовление список для отделов лаборатории;
- непосредственный ввод данных измерений в гросбух;
- ввод данных измерений в гросбух по отделам лаборатории.

ФУНКЦИИ В ПОЛДЕНЬ:

- печат данных осмотра.

ФУНКЦИИ ВЕЧЕРОМ:

- оценка и обработка данных хематологического автомата;
- печат осмотров данных (в порядке отделов госпиталя);
- печат статистики.

ПРОЧИЕ ФУНКЦИИ:

- приготовление статистик;
- качественный контроль.

СИСТЕМА НА ОТДЕЛЕ ГОСПИТАЛЯ

Регистрация больных на детском отделе:

Обычный приём данных дополняя с данными анамнеза.

Средства описания положения больных:

- хранение текстовых информаций в кодированном форме, их декод;
- приготовление данных осмотра (вычисление, печат (гемодинамический диаграмм));

Прочие элементы системы:

- разработка данных больных по сложным логическим условиям;
- следование больному, амбулантная система;
- база данных для работы врачей (регистр медицинских средств).

СИСТЕМА АВТОМАТИЗИРОВАННОГО СОСТАВЛЕНИЯ СВОБОДНЫХ ТЕКСТОВ

Проф. П. Барнев - До Вьет Нга

I. ВВЕДЕНИЕ

Вопрос автоматизации работы с документацией с помощью компьютеров является одним из направлений применения вычислительной техники в административной деятельности. Создан ряд текстообработывающих систем /или как часто говорят - текстовые редакторы/, упрощающие подготовку текстов. В данной работе предлагается одна модель текстообработывающей системы, где применяются приемы для повышения степени автоматизации процесса обработки. Эти приемы основаны на эффективные средства для использования других документов /или так называемой в данной работе текстовых материалов/ и для автоматизации процесса ввода текста с помощью словаря.

Предлагаемая модель редактора посвящается прежде всего пользователям, работающим с текстами нестандартного типа, особенно в творческом процессе. Модель является удобным средством также для работы служащего типа секретаря. В работе рассматриваются еще проблемы, возникающие при реализации модели.

II. ПРОЦЕСС КОМПОЗИЦИИ ДОКУМЕНТОВ

С развитием ЭВМ и быстрым снижением их цены вполне возможно думать о перспективной возможности персонального наличия компьютера как инструмента в творческой работе. Всем, кто работает с документацией, хотелось бы иметь компьютерные средства, упрощающие его труд.

Рассмотрим сначала процесс создания литературного документа и возможные проблемы. Примером такого процесса может служить работа научного работника над некоторой научной ста-

твей. В своей работе научный работник пользуется текстовыми материалами, включая также и другие его работы, и он работает в тесной связи с его коллегами и руководителем. Процедура его работы следующая:

P1. Изучение имеющихся документов.

P2. Составление плана содержания статьи.

P3. Работа над пунктами статьи, требующая выполнения следующих действий:

- выбор подходящих участков из текстовых материалов;
- ввод своих фрагментов;
- упорядочения фрагментов и необходимой последовательности;
- исправление текста.

P4. Создание окончательного документа.

P5. Обсуждение документа с коллегами и руководителем.

P6. Исправление и изменение документа.

Процедура сама по себе трудоемкая, более того часто требуется повторить много раз. С помощью существующих в настоящее время текстообработывающих средств-редакторов облегчается в некоторой степени работа. Для дальнейшего развития уровня автоматизации, ниже предлагаются два метода обработки.

III. МЕТОД ОБРАБОТКИ

1. Общее описание

Обработка текстовых материалов является неразделимой частью в процессе создания документа. Правильное и быстрое ориентирование в литературе не только ускоряет процесс составления документа, но и улучшает его качество как научное произведение. В рассматриваемой выше схеме работы работника процедура P1 - P4 представляет собой процесс редактирования и обработки текстовых материалов. На практике она является наиболее трудоемкой. Но часто именно она определяет качество работы, делает статью более ясной, более

наглядной и более доступной для понимания. Облегчение процедуры редактирования имеет значение не только для работника, как человека, работающего с письменным материалом, но может и повлиять на качество содержания работы.

При реальной возможности можно допустить, что составитель имеет в своем распоряжении персональный компьютер, в котором содержатся необходимые ему текстовые материалы. Нижеописанный метод предполагает один способ хранения и обработки текстовых материалов на компьютере.

Текстовые материалы разделяются на документы. Последние в свою очередь разделяются на иерархию пунктов. Компьютер кроме текста документов хранит еще дерево иерархии пунктов, в котором содержится краткая необходимая информация о структуре и содержании документа.

С помощью такой организации компьютер эффективно помогает пользователю в быстром рассмотрении содержания документов, что ускоряет процесс ориентирования и усваивания наличия литературы. Кроме того, пользователь легко может обратиться к желаемому ему участку документа и фиксировать его в обрабатываемом документе.

Для получения документа в вышеописанном виде от пользователя не требуется особого труда. Необходимо только правильно ввести в текст документа цифровую номерацию /аналогичную номерации, использованной в данной работе/, т.е. документ разделяется на пункты

- 1.
- 2.
3. ...

пункт 1 разделяется на пункты

- 1.1
- 1.2 ...

пункт 2 разделяется на пункты

- 2.1
- 2.2 ...

...
пункт p1, p2, ..., pk разделяется на пункты
p1.p2 ... pk1
p1,p2 ... pk.2
...

После номера пойдет титр пункта. Здесь титр одновременно играет роль краткой информации о пункте. В случае отсутствия титра первая строка пункта воспринимается как краткая информация о данном пункте. Компьютер при первом рассмотрении документа фиксирует дерево иерархии и необходимую информацию.

В процессе создания письменных работ часто требуется переработка документа, причем много раз. При переработке содержание документа уже известно, оно содержится в старом экземпляре и его надо только по-другому упорядочить, изменяя текст в некоторых местах. Для такой обработки можно облегчить работу включением старого экземпляра в набор текстовых материалов. Такой прием можно применять также для создания документа, большая часть содержания которого находится в другом документе. Например, создание сообщения о результате заседания совета, когда имеется протокол заседания.

При наличии такого средства обработки компьютер может активно участвовать в творческой деятельности научного работника.

2. Проектирование метода

2.1. Хранение текстовых материалов

Текстовыми материалами назовем набор имеющихся в системе документов. Их список хранится в каталоге текстовых материалов. Текст каждого документа разделяется на иерархию пунктов, которая имеет структуру дерева и может быть выражена графом следующего вида:

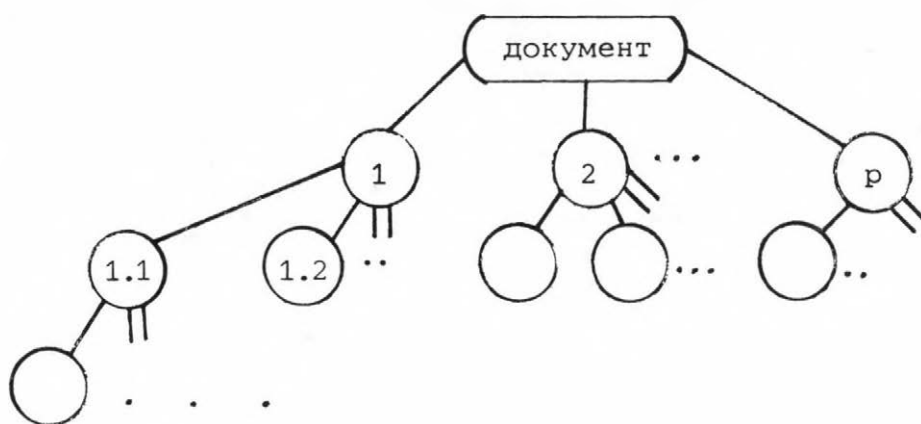


Рис. 1.

Во внутреннем представлении каждый документ текстовых материалов хранится в двух файлах: файл текста документа и файл структуры документа /ФТ и ФС/. ФС, хранящий структуру документа, состоит из записей, каждая из которых содержит информацию о соответствующем пункте. Записи в ФС находятся в следующем порядке: первая запись хранит только титр документа. Каждая запись выражает один узел в дереве структуры, поэтому соответственно имеет своих братьев и наследников. Записи-братья хранятся последовательно по возрастанию номера записи. Например:

p1.p2 ... pk.1	<input type="text"/>
p1.p2 ... pk.2	<input type="text"/>
p1.p2 ... pk.3	<input type="text"/>
p1.p2 ... pk.4	<input type="text"/>

В каждой записи указывается ее первый наследник и число ее прямых наследников. Таким образом граф на рис. 1 в файловом представлении имеет вид:

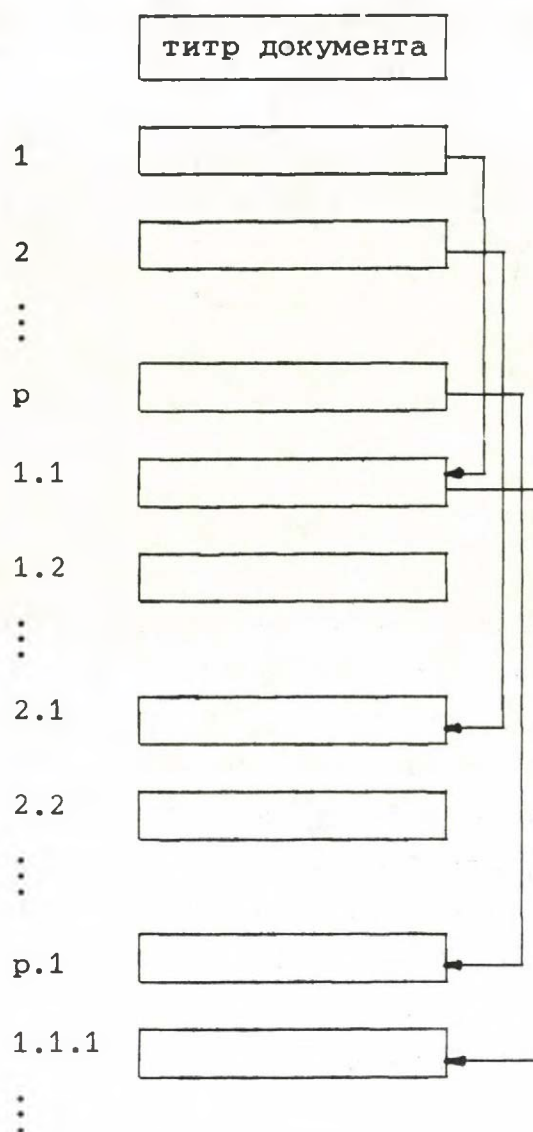


Рис. 2.

Одна запись состоит из 4-х частей информации:

- 1/ Краткое содержание пункта: однострочная фаза текста.
- 2/ Число прямых наследников: число.
- 3/ Указатель первого наследника в ФС: число.
- 4/ Место нахождения текста пункта в ФТ: номер страницы, номер строки на странице.

2.2. Ввод нового документа в набор текстовых материалов

Для ввода нового документа в набор текстовых материалов

требуется только наличие в компьютере его текста с цифровой нумерацией, описанной в пункте 2.1. По объявляемому названию система сама анализирует файл текста документа ФТ, в результате чего оформляется файл структуры документа ФС. Этот процесс выполняется в двух фазах:

I-ая фаза: Фаза анализа ФТ. Система читает ФТ выделяя в специальном буфере в соответствии с каждым пунктом его номер и титр.

II-ая фаза: На основе информации в буфере система упорядочит записи по нужному порядку и вычисляет соответствующие указатели. Таким образом образуется ФС и документ готов для использования.

2.3. Использование набора текстовых материалов

Для использования набора текстовых материалов пользователь вызывает систему обработки материалов, которая предоставляет ему список документов и по его приказу начинает работать с конкретным документом. Пользователь может читать документ по содержанию или по тексту, может также читать только отдельный пункт документа. Но особым средством для обработки текстовых материалов в данной работе является возможность использования материалов в процессе создания нового документа. Т.е. пользователь может подключать к работе систему обработки материалов параллельно с работой редакторской системы. При этом пользователь имеет не только возможность чтения литературы, но и переписки необходимого ему участка в своем документе.

IV. МЕТОД АВТОМАТИЗАЦИИ ВВОДА ТЕКСТА

1. Общее описание

Работа с документами никогда не обойдется без ввода текстов. Особенно когда дело касается творческой работы - работы с документами нестандартного типа. Поэтому эффективный прием для ускорения ввода текстов может прямо повлиять

на скорость и надежность работы. Ниже предлагается один метод ввода текстов с помощью словаря. Идея метода следующая:

В обыкновенном редакторе, удобном для исправления и ввода текстов, включается средство для ускорения ввода текстов с помощью так называемого "словаря". Словарь является списком наиболее использованных в данной работе слов, подготавливаемым самим составителем документа. В процессе ввода или даже исправления текста составитель в место ввода целого слова или словосочетания, имеющегося в списке, вводит только первые их буквы. Например, в списке имеется словосочетание:

"Институт математики с ВЦ"

тогда во время ввода текста в место этого длинного сочетания составитель вводит только следующую группу знаков

"и&"

где "&" - знак использования словаря

"и" - первая буква словосочетания.

Такой метод использования словаря дает возможность сочетания с использованием обыкновенного метода ввода текста с клавиатуры. Т.е. составитель работает по традиционному способу ввода текста с клавиатуры, при этом вводя слова или словосочетания, о которых он уже "договорился с системой", он имеет право вводить только их сокращения. Такой метод "Сокращения" не является чем-то новым. В повседневном тексте часто встречается, например, сочетание "и т.д.", что означает "и так далее". Где "т" и "д" являются первыми буквами слова, а "." играет роль знака "&", т.е. указывает, что в данном случае идет сокращение.

Если обратить внимание на обстоятельство, что обычно люди работают в определенной сфере деятельности, т.е. работают с текстами, где часто встречаются определенные многократно повторяющиеся слова и словосочетания, то использование вышеописанного метода действительно может повысить скорость ввода текста.

Например, для выше двухстраничного текста пункта 1 в качестве словаря наиболее использованных слов можно выбрать следующий список:

метод	котор
автоматизаци	помощ
ввод	слов
текст	словосочетани
работ	составител
документ	использовани
скорост	возможност
словар	клавиатур
следующ	систем
исправлени	сокращени
ускорени	определенн

При использовании вышеописанного метода с вышепоказанным словарем время ввода может сократиться на 30 %.

2. Проектирование метода - редактор со словарем

В данном пункте будем анализировать работу редактора со словарем. Как выше уже сказано, использование словаря в процессе ввода текста реализуется как одна из возможностей редактора. Не будем рассматривать все обыкновенные редакторские функции, такие как: сдвиг курсора в различные направления, стирание, добавление и ввод текста. Подробно будем останавливаться только на функции, связанной с работой со словарем.

Использование словаря допускается везде, где имеется ввод текста с клавиатуры. Оно объявляется через знак использования словаря /или иначе говоря, знак сокращения/ "&". При встрече знака "&" редактор выделяет группу первых букв сокращенного слова, по этой группе находит полное слово в словаре и запишет его в текст.

2.1. Выделение сокращенного слова

Группа первых букв сокращенного слова стоит между последним пробелом и знаком "&".

Примеры:

1 1 2 3 &
 □ □ □ □

2 и так д.

3 и такд.

На первом примере этой группы является группа, находящаяся между 3-им пробелом и знаком "&". На втором примере - "д". А на третьем примере - "такд".

В этом процессе важно отметить, что последний пробел играет роль разделителя, а знак "&" играет роль и знака сокращения и разделителя. Использование пробела как разделителя дает возможность сэкономить одно действие: ввод первого разделителя сокращения.

2.2. Поиск слова в словаре

Когда уже выделена группа первых букв сокращения система приступает к поиску самого слова в словаре. Чтобы упростить работу пользователя, избавить его от необходимости помнить словарь, в системе предусмотрены 3 случая:

- в словаре имеется только одно слово, начинающееся с данной группы : слово вводится сразу в текст;
- в словаре имеются несколько слов, начинающиеся с этой группы: выводится на экран подсписок этих слов и предоставляется пользователю возможность выбора;
- в словаре нет слова, начинающегося с данной группы: система вернется к положению знака сокращения "&" и ожидает окончания с клавиатуры. При этом предоставляется возможность ввода данного слова в словарь.

2.3. Ввод слова в словарь

Словарь - список возможных сокращений слов и словосочетаний можно заранее подготовить и ввести в систему через редактор. Для более удобного использования предусмотрена и возможность ввода слова в словарь в процессе работы с текстом. Таким образом даже по ходу работы пользователь может "договориться" с системой о своем сокращении. Эта возможность реализуется следующим образом: например, после ввода в текст некоторого слова или словосочетания, пользователь желает его ввести в словарь, он вводит знак ввода в словарь "!" или "!<число>". Случай "!" означает, что пользователь желает включить последнее введенное слово в словарь, а случай "!3" означает - последнее введенное словосочетание, состоящее из трех слов. Здесь система понимает под словом группу знаков, находящихся между двумя разделителями: пробел, запятая, точка и др.

В использовании словаря не предусматривается проблема, возникающая с изменением окончания слов в некоторых языках, как например, в русском языке слово "русский" может появиться в тексте как "русский", "русская", "русское", "русского" и т.д. Для таких слов вводится в словаре только его корень - "русск" /точнее говоря, неизменная часть слова/ окончание приходится добавить с клавиатуры.

V. МОДЕЛЬ РАБОТЫ СИСТЕМЫ

Система имеет две главные части, хорошо описанные в предыдущих пунктах: часть обработки текстовых материалов и редакторская часть. Пользователь может работать с любой частью как с самостоятельной системой. Но главным преимуществом системы является возможность параллельной работы двух частей, т.е. имеется возможность работая в редакторской части подключить и работу с текстовыми материалами. Таким образом пользователь может обработать свой материал заглядывая, используя, копируя другие готовые материалы.

Такая возможность реализуется с помощью вспомогательного экрана, который вызывается нажатием определенной фиксированной клавиши. Такая модель работы является аналогом обычного процесса создания литературной работы. При этом работник располагает эффективным средством для использования текстовых материалов и удобным ускоряющим средством для ввода и исправления текста.

VI. ЗАКЛЮЧЕНИЕ

Для автоматизации работы с документацией были сделаны много попыток. Были созданы текстовые редакторы, облегчающие внесение изменений и форматирование текста. Были также разработаны системы, упорядочивающие процесс ввода текста стандартного типа с помощью заранее введенных макетов, бланков. Описанная в данной работе модель системы ускоряет сам процесс ввода текста, что дает автоматизированное средство для составления текста нестандартного типа. Более того, она также предлагает эффективное средство для обработки содержания документов, что является необходимым для творческого процесса.

Подготовка документов с помощью этой модели дает существенное ускорение работы, уменьшение пропусков и неточностей, возможность дальнейшей компьютерной обработки документов и учета проводимых работ.

Система может быть реализована на микрокомпьютерах, что является доступным для большого круга пользователей. Языком документов и языком общения с системой может быть любой естественный язык, использующий алфавит, допустимый в компьютере.

Работа по усовершенствованию и реализации модели продолжается.

Текстовая база данных

А. Б. Борковский
(ВЦ АН СССР, Москва)

A text database NOTES for IBM PC provides storing, retrieval and editing of text records. Each record can have arbitrary format and its length is unlimited. The records are accessed by keywords with "many-to-many" relation between records and keywords. The desktop-like interface permits scrolling through keywords list or the list of records titles and working with several records at the same time. The package includes full-featured screen text editor to read, modify and create text record.

1. "Интеллектуализация" ЭВМ должна обеспечивать работу с машины с неподготовленным пользователем. Можно выделить два аспекта "интеллектуализации":

- с одной стороны, применение баз знаний позволяет автоматически выполнять сложные действия, не требуя от пользователя написания программ или подробного задания алгоритмов;
- с другой стороны, развитые средства

непосредственного взаимодействия обеспечивают удобную работу с большими объемами информации без знания языков манипулирования данными и, предоставляя множество мелких удобств, позволяют пользователю "в ручную" достигать цели.

Описанная ниже текстовая база данных относится к средствам второго рода.

2. Что такое текстовая база данных

2.1. Текстовые базы данных оперируют с плохо структурированной информацией, элементы которой неудобно представлять в табличном виде, как это принято в обычных базах данных. Обычные базы данных ориентированы, в первую очередь, на выполнение массовых операций - сортировку, поиск, подготовку отчетов и обработку прикладными программами; к ним обращается организованная группа пользователей; за структурой и содержанием базы данных следит администратор базы данных.

Текстовые базы данных предназначены для интерактивной работы с разнородной информацией и ориентированы на пользователя с минимальной подготовкой. Поэтому работа с ней опирается на удобство "ручной" работы с данными, а не на

составление сложных запросов. Все операции текстовой базы выполняются с помощью управляющих клавиш или меню, так что пользователю не надо изучать специальный язык.

2.2. Текстовая база данных представляет собой совокупность КАРТОЧЕК, содержащих текстовую информацию. Тексты карточек имеют произвольный формат, они не обязаны быть похожими друг на друга. В каждой карточке выделяется образом одно или несколько КЛЮЧЕВЫХ СЛОВ, обеспечивающий поиск и доступ к карточкам. Один и тот же ключ может входить в несколько карточек. Отсюда другое название для текстовых баз данных - картотека.

В работе с разработанной нами для ПЭВМ типа IBM PC картотеккой NOTES можно выделить следующие характерные операции:

- просмотр списка ключевых слов
- поиск ключа, по заданному шаблону
- "раскрытие" ключа, - показ первых строчек карточек
- просмотр и редактирование текста выбранной карточки
- занесение отредактированной карточки в картотеку
- "откладывание" просмотренной карточки
- установка шаблона, определяющего подмножество ключевых слов для работы
- создание новой карточки и занесение ее в

картотеку

- вывод одной, нескольких или всех карточек в текстовых файл или на печать
- ввод в картотеку новых карточек из текстового файла
- удаление карточек и/или ключевых слов
- выполнение операций операционной системы

Выбранные карточки просматриваются и редактируются в отдельных окнах (можно иметь до девяти рабочих окон одновременно). Встроенный в картотеку редактор текстов обеспечивает все традиционные операции экранного редактора - просмотр, вставку и удаление символов, строк и выделенных фрагментов, контекстный поиск, работу с двумя (русским и латинским) алфавитами и пр. Ключевые слова в тексте выделяются цветом.

2.3. Картотека NOTES реализована на основе Администратора Окон. Каждый упорядоченный набор объектов (список ключей, строки текста карточки, меню) отображается в отдельном окне. Работа с любым окном производится единообразно: клавиши управления курсором обеспечивают прокрутку и выбор текущего объекта, нажатие функциональной клавиши вызывает соответствующее действие над текущим (указываемым курсором) объектом. При этом аналогичные действия (удаление, раскрытие, запись в файл, подсказка) вызываются одинаковыми клавишами во всех окнах. Администратор окон позволяет

также изменять размер и расположение окон на экране; установленные параметры окон запоминаются и используются при последующих вызовах картотеки.

3. Работа с картотекой NOTES

3.1. Работа с картотекой NOTES основана на непосредственном взаимодействии: для выполнения операции пользователь выбирает необходимый объект клавишами управления курсором и нажимает функциональную клавишу. Строка в нижней части экрана содержит краткую подсказку о возможных в данный момент действиях; полнее полная инструкция вызывается клавишей <подсказка>.

3.2. Работа со списком ключевых слов.

После вызова картотеки на экране изображено окно со алфавитно-упорядоченным списком ключевых слов. Вы можете просматривать этот список, перемещаясь по нему клавишами управления курсором, или сразу найти необходимый ключ, нажав клавишу <поиск> и задав ключ; поиск производится по первым буквам. Команда <поиск> используется также для выбора группы карточек.

Выбрав в окне ключевых слов интересующее Вас слово, Вы можете (а) раскрыть его, (б) посмотреть связанную с ним карточку или (в) напечатать или записать в файл карточки с этим ключевым словом.

Для раскрытия надо нажать клавишу <раскрыть>; при

этом изображение слова заменяется списком заголовков (первых строчек) карточек, в которых оно встречается. Повторное нажатие клавиши <раскрыть> возвращает изображение только ключевого слова.

Указав на ключевое слово или на заголовок карточки в окне ключевых слов можно вызвать всю карточку в рабочее окно клавишей <взять карточку>. Имеется несколько рабочих окон; цифра перед клавишей <взять карточку> указывает номер окна, в которое помещается карточка. Клавиши **alt**<цифра> позволяют перемещаться между окнами; при переходе в другое окно содержимое текущего окна не изменяется.

Перейти в рабочее окно можно также командой <создать>. При этом Вы начинаете вводить новую карточку в пустом рабочем окне.

Ключи_____		
Алексеев		
Мячев		
* Николаев	1	
Орлов	Николаев Юрий Петрович, старший инженер	
Пономарев	род. 10/11/1958	
Рыжов	образов. высшее, оконч. МФТИ в 1983г.	
Тарасов	принят в ВЦ 01/09/1983	
Якушев	з/п 220	

Рис.1. Вид экрана при выводе списка ключей и одной карточки

3.2. Операции в рабочем окне.

В рабочем окне с текстом карточки можно работать как в обычном экранном редакторе: перемещаться по тексту и прокручивать (**scroll**) его, вставлять, удалять (**cut**), восстанавливать (**paste**) и копировать литеры и строки. Текст, удаленный из карточки в одном рабочем окне можно вставить в другом окне. Ключевые слова выделяются в тексте цветом; при редактировании строки ключевые слова отмечаются управляющим символом, который может быть удален как обычная литера и вставлен клавишей <отметить ключ>. Отредактированная карточка может быть записана клавишей <записать>. При этом, однако, список ключевых слов не изменяется.

Для того, что бы занести в картотеку ссылки на измененную или созданную карточку, необходимо вызвать клавишей <изменить> меню модификации (см. ниже).

Клавиша <выход> возвращает Вас в окно ключевых слов.

3.3. Добавление и удаление карточек.

Карточки можно удалять клавишей <удалить> в окне ключевых слов и с помощью меню модификации, вызываемого клавишей <изменить> в рабочем окне.

Вы можете добавлять карточки в картотеку одним из трех способов:

ввод карточки в пустое рабочее окно клавишей

<создать>;

модификация имеющейся карточки;

ввод карточек из текстового файла; см. ниже.

3.4. Клавиша <удалить> уничтожает связь ключевого слова и карточки. Если курсор указывает на заголовок карточки при "раскрытии" ключевого слова, уничтожается ссылка на эту карточку; если же курсор указывает на ключевое слово, уничтожаются все ссылки этого ключевого слова. Карточки, на которые нет ссылок, удаляются из картотеки.

3.5. Меню модификации - добавление и удаление карточек.

Меню модификации содержит три пункта:

занести новую - карточка в текущем рабочем окне заносится в картотеку;

занести старую - в списке ключевых слов убираются все ссылки на карточку в текущем окне и карточка заново заносится в картотеку;

удалить - карточка в текущем окне удаляется из картотеки.

Операция "занести старую" фактически эквивалентна последовательному выполнению операций "удалить" и "занести новую".

3.6. Главное меню.

В главное меню Вы попадаете, выйдя из окна ключевых слов клавишей <выход>. Главное меню содержит три альтернативы:

ключи - переход в окно ключевых слов

шаблон - задание шаблона для ключевых слов,

показываемых в окне ключевых слов; например шаблон "пет* фамилия, н*" выделяет карточки, содержащие либо ключевое слово, начинающееся на букву "н", либо содержащие ключевое слово "фамилия" и ключевое слово, начинающееся на "пет".

ввод/вывод - вызывает меню ввода/вывода.

3.7. Меню ввода/вывода.

Меню ввода/вывода содержит следующие альтернативы:

Сохранить все - содержимое картотеки выводится в текстовый файл. При этом каждая карточка оформляется как абзац - последовательность строк, первая из которых начинается пробелом, а остальные начинаются отличной от пробела литерой.

Сохранить часть - карточки, удовлетворяющие заданному шаблону выводятся в текстовый файл.

Имя файла - задать имя файла, в который выводятся карточки клавишей <записать>.

Загрузить - ввести карточки из текстового файла.

Каждый абзац заносится как отдельная карточка; ключевые слова должны быть отмечены кодом ASCII 11 (ctrl_K).

Создать - текущая картотека очищается и в нее загружается указанный текстовый файл.

4. Реализация и использование.

Картотека **NOTES** написана на **TURBO-Pascal'**е. Настоящая версия является прототипом соответствующей компоненты интегрированной системы **СПЕКТР**, разрабатываемой в Лаборатории программного обеспечения микроЭВМ ВЦ АН СССР.

Картотека **NOTES** находится в стадии опытной эксплуатации. Ее используют, в основном, в качестве записной книжки для хранения разнообразной текущей информации, для хранения библиографических ссылок и др. Автор использовал картотеку **NOTES** для подготовки к изданию англо-русского словаря по информатике и программированию с толкованиями. Объем словаря - около 7000 словарных статей, более 15000 входов в списке ключевых слов, общий объем текста карточек - более 800К байт.

5. Соответствие команд NOTES и клавиш для IBM PC

5.1. Команды списка ключевых слов:

<подсказка>	- F1, shift_F1
<выход>	- F10
<поиск>	- F2
<удалить>	- ctrl_F3
<создать>	- F7
<записать>	- shift_F8
<раскрыть>	- Grey Plus
<взять карточку>	- Enter
<изменить окно>	- alt_W
<переключение окон>	- alt + цифра

5.2. Команды рабочего окна (кроме команд редактирования):

<записать>	- shift_F10
<изменить>	- ctrl_F10
<отложить>	- F8
<след. карточка>	- ctrl_PgDn
<пред. карточка>	- ctrl_PgUp
<отметить ключ>	- ctrl_K
<поиск текущего слова>	- shift_F2
<изменить окно>	- alt_W
<переключение окон>	- alt + цифра

Интеграция баз данных и баз знаний

Г. Виткова, Д. Гусек - Центральный вычислительный институт
ЧСАН

1. Введение

Впервые понятие банка данных было сформулировано в 1965 г. С. Новоботтером, который в это время руководил работами по созданию банка данных для шведского статистического управления. Первые работы в области искусственного интеллекта (далее ИИ) появились уже в пятидесятых годах. Тем не менее развитие обеих областей тесно связано с бурным развитием вычислительной техники в течение последних двадцати лет. До самого последнего времени обе эти области развивались независимо одна от другой.

Оказалось, что совместное использование результатов исследований в области ИИ (в первую очередь создание так называемых баз знаний) и результатов теоретических разработок в области системы баз данных (далее СВД) могут принести серьезные выгоды с практической точки зрения.

В последние годы в рамках технологии СВД было создано много эффективных систем для хранения и поиска больших и сверхбольших массивов данных. В настоящее время существует большое количество практически реализованных и хорошо работающих, функционирующих СВД, основанных на реляционной, иерархической, сетевой моделях и их модификациях. Эти системы установлены как на крупных ЭВМ, так и на малых мини- и микро ЭВМ, а кроме того во всем мире ведутся интенсивные разработки по созданию процессоров баз данных.

В области развития ИИ в последние годы созданы успешные экспертные системы, в которых накапливаются и используются знания, необходимые для решения определенных проблем. Составной частью разработок в этой области являются работы по созданию методов, позволяющих создавать так называемые дедуктивные механизмы, которые поддерживают построение и использование баз

Несмотря на достигнутые успехи обе рассматриваемые технологии имеют свои ограничения, которые препятствуют в случае их изолированного применения существенному повышению их эффективности.

Существующие экспертные системы, имеющие дело лишь с сотнями или тысячами единиц знаний, не имеют средств для эффективного манипулирования с большими массивами данных. Существующие СВД, умеющие напротив тому обеспечить высокоэффективное манипулирование данными, не умеют эффективно работать с данными в виде правил, которые встречаются в экспертных системах.

Объединяющей основой теоретических исследований и практических применений в обеих областях стала математическая логика [3 , 6] , которая является мощным инструментом в виде языков так называемого логического программирования.

Имея в виду вышесказанное, можно считать, что данные, хранимые в базе данных, представляют конкретные факты, характеризующие конкретные явления окружающего реального мира. Знания же, находящиеся в базе знаний, представляют из себя обобщенные факты о наших представлениях о реальном мире (см. рис. 1).

Сама логика формализована в понятиях теории доказывания и теории моделей. Теория доказывания используется для генерирования ответов на пользовательские запросы путем постепенных построений следствий на основе имеющихся данных. Теория моделей предоставляет средства для решения вопроса о том, каким образом можно утверждения применить для обработки массива конкретных данных.

Дополнение СВД дедуктивным механизмом приносит много выгод, главными из которых являются:

- 1) Возможность выбирать информацию, которая не находится в базе данных в явном виде, однако может быть получена путем логического вывода на основе хранимых данных.

- 2) Поддержка создания и использования непроцедуральных так называемых пользовательских языков.
- 3) Возможность получить вместе с ответами пояснение доводов за и против предлагаемого ответа. Сам процесс вывода следствий является точной, строгой математической операцией. Однако исходные предпосылки основаны как правило на неопределенных данных о реальном свете (тем более это справедливо в случае научных и экспериментальных данных). Поэтому ответы, в которых дается мера приемлемости ответа, очень часто более ценны, чем ответы типа "да" или "нет".
- 4) Возможность ответа на запросы типа "что - если" и на другие подобные вопросы, которые практически нельзя задать существующим СВД. Кроме того дедуктивные механизмы помогают соблюдать ограничения целостности базы данных.

2. Предполагаемая область использования

Исследования, проводящиеся в Институте физиологических регуляций ЧСАН (далее ИФР ЧСАН), с которым сотрудничает Центральный вычислительный институт ЧСАН (далее ЦВИ ЧСАН), нуждаются в интегрированной системе для поддержки создания моделей процесса учения, которая бы обладала достоинствами СВД и систем ИИ.

Эксперименты по исследованию процесса учения в ИФР ЧСАН выглядят следующим образом. Участники эксперимента подвергаются воздействию звуковых и световых сигналов. При этом наблюдается реакция организма на эти сигналы в виде следующих регистрационных записей:

- а) миограмм, т.е. записей, регистрирующих влияние сигнала на мускулы века,
- б) электроэнцефалограмм (далее ЭЭГ),
- в) электрокардиограмм (далее ЭКГ),
- г) плетизмограмм, т.е. записей, регистрирующих изменения кровяного давления на кончиках пальцев.

Через некоторое время прекращается подача светового сигнала (остается только звуковой), и снова проводятся наблюдения тех же самых физиологических величин.

Целью исследований прежде всего является:

- 1) нахождения параметров, характеризующих затормаживание и возбуждение физиологической системы в процессе учения;
- 2) исследование значения латентного периода и сопутствующих физиологических явлений;
- 3) исследование динамики появления связи;
- 4) классификация людей в зависимости от динамики появления связи;
- 5) определение биологического ритма высшей нервной деятельности человека;
- 6) создание моделей процесса учения.

На основе имеющегося опыта анализа и обработки экспериментальных данных с помощью ЭВМ [2 , 7] можно сформулировать следующие замечания.

Для обработки экспериментальных данных с помощью ЭВМ проводится дигитализация регистрационных записей, в результате чего возникают обширные массивы данных, эффективное манипулирование которыми невозможно без специальных, предназначенных для этой цели средств.

Анализ каждой физиологической величины имеет свои специфические особенности. Поэтому методы и программы, используемые для обработки и анализа отдельных физиологических величин будут отличаться и отличаются друг от друга. По этой причине целесообразно накапливать знания о программах, наиболее подходящих в конкретных случаях для отдельных физиологических величин.

Собственно анализ данных направлен на поиск и определения закономерностей как внутри физиологических величин, так и между ними. Полученные закономерности тоже целесообразно систематически накапливать.

Модели процесса учения будут создаваться постепенно на основе частных, ранее определенных закономерностей. Модели необходимо проверять и уточнять, используя массивы экспериментальных данных. Знания о моделях и результатах их апробирования тоже целесообразно собирать и накапливать.

Из вышесказанного вытекает два важных вывода. Во-первых, в рамках научных исследований и экспериментов возникают большие массивы данных. Так например, в результате одного эксперимента, поставленного с целью изучения механизма учения, возникают данные в объеме, измеряемом мегабайтами. Предполагаемое количество экспериментов по отдельной тематике измеряется сотнями. В результате обработки первичных экспериментальных данных возникают новые массивы данных, некоторые из них заслуживают длительного хранения. Между величинами, полученными в результате расчетов на ЭВМ, необходимо также искать закономерности. В процессе анализа необходимо снова и снова возвращаться к старым данным и снова их анализировать с позиций новых, полученных в результате исследований закономерностей. Во-вторых, в процессе исследований постепенно накапливаются знания о наиболее подходящих для анализа программах, о полученных закономерностях в исследуемом физиологическом объекте, о моделях процесса учения. Все эти знания необходимо хранить и на их основе выводить более общие, глобальные закономерности. В ходе анализа эксперимента бывает необходимо исследовать огромное количество возможных взаимосвязей, что без автоматизации нельзя практически сделать.

Отсюда вытекает необходимость в создании интегрированной системы для поддержки всего процесса научных исследований, главные функции которой можно сформулировать следующим образом:

- 1) хранение и накопление экспериментальных и научных данных в избранной области исследований;

- 2) сбор и хранение информации о наиболее подходящих для данных исследований программах, постепенное создание проблемно-ориентированных библиотек алгоритмов и программ;
- 3) хранение и накопление знаний об исследуемых явлениях, о моделях, отображающих эти явления, включая возможность автоматизированного генерирования гипотез о закономерностях исследованного процесса на основе хранимых данных.

На основе вышеизложенного можно сделать общее заключение о том, что в описываемой области физиологических исследований и в подобных ей областях исследования созрели условия для создания интегрированной системы, использующей достоинства и возможности как СВД, так и систем ИИ с одновременным использованием и мобилизацией всех возможностей имеющейся операционной систем, в среде которой предполагается работа интегрированной системы.

3. Существующий способ использования ЭВМ для поддержки анализа экспериментальных данных

В настоящее время в ЦВИ ЧСАН для анализа экспериментальных данных используется программное обеспечение из довольно богатого фонда алгоритмов и программ ЧСАН /далее ФАП ЧСАН/, физически находящегося в ЦВИ ЧСАН. При использовании программ из ФАПа имеются следующие трудности/рис. 2/:

- Ввиду большого объема ФАПа не всегда ясно, какие программы наиболее подходят для рассматриваемого случая. Например, априори можно сказать, что для анализа ЭЭГ и миограмм будут нужны разные программы, но уже сложно рекомендовать программы для обработки миограмм, т.е. отсутствуют систематические знания о наиболее подходящих для данной физиологической величины программах.
- Сам процесс использования программ медленный и дискомфортный. Исследователь для того, чтобы выбрать необходимые программы

должен изучить толстые тома документации, а потом составить главную программу.

- В случае применения базы данных исследователю нужна помощь, иногда весьма существенная, специалиста по базам данных.

Использование DMS для поддержки анализа экспериментальных данных началось в ЦВИ ЧСАН сравнительно недавно [7].

В ходе применения DMS необходимо провести следующую работу:

- 1/ спроектировать логическую структуру базы экспериментальных данных,
- 2/ в соответствии с логической структурой провести определение базы данных, применяя язык определений используемой DMS
- 3/ загрузить определенные в п.2 данные в базу данных.

После этого можно:

- а/ производить поиск и выборку данных в структуре, требуемой для выбранных программных средств / предпочтительнее с помощью непроцедурального языка/,
- б/ осуществлять в случае необходимости реструктуризацию базы данных в зависимости от способа обработки экспериментальных данных.

Для поддержки логического проектирования базы экспериментальных данных только что начаты разработки соответствующих программных средств. Пока отсутствует системная поддержка диалогового способа работы как с базой данных, так и с ФАПом.

Краткая характеристика существующего положения дел показывает, что поддержка анализа экспериментальных данных на ЭВМ в настоящее время явно недостаточна и что создание интегрированной системы может привести к серьезному улучшению положения в этой области.

4. Перспективы развития интегрированной системы

Как уже было сказано, для эффективной поддержки анализа экспериментальных данных необходимо создать интегрированную систему, основными составляющими которой могут быть:

- а) система управления базой данных вместе с базой данных;
- б) система управления программами вместе с фондом программ;
- в) система управления базой знаний вместе с базой знаний.

Работы в этом направлении предполагается проводить по этапам (с учетом того, что уже имеется определенный опыт по использованию DMS для экспериментальных данных).

I этап /рис. 3а/ - создание информационной системы о программах, в которую будет включено:

- а) описание программ;
- б) знания о целесообразности применения отдельных программ в конкретных случаях;
- в) знания о требованиях операционной системы, в среде которой работают программы, и требования к структуре входных и выходных данных.

Предполагается, что знания о программах будут включаться в базу знаний непосредственно экспериментатором.

II этап /рис. 3б/ - информационная система I этапа дополнится управляющими элементами и интерфейсом с DMS, в результате чего эта система сможет выполнять следующие функции:

- а) поможет пользователю в диалоговом режиме выбрать и запустить программы в соответствии с его требованиями;
- б) одновременно поможет выбрать соответствующие данные из базы данных, а в случае необходимости включить результаты расчетов в базу данных.

III этап - постепенное создание базы знаний, в которой предполагается различать три уровня /рис. 4, 5/:

- знания о программах;
- знания о полученных закономерностях в исследуемых физиологических процессах;
- знания о моделях процесса учения.

Знания о программах будут включаться в базу знаний на основе полученного опыта прямо экспериментатором.

Знания о закономерностях в исследуемых процессах предполагается получать в первую очередь с помощью программной системы для генерирования гипотез, о которой чуть позже расскажем подробнее. Остальные знания о закономерностях могут включаться в базу знаний по рекомендации экспериментатора.

Знания о моделях будут включаться в базу знаний непосредственно экспериментатором.

Остановимся подробнее на выводе закономерностей на основе экспериментальных данных, точнее их характеристик, находящихся в базе данных с помощью системы GUNA.

Эта система была разработана в ЧСАН и служит для генерирования гипотез об исследуемых объектах. Система предназначена для работы в первую очередь с дихотомическими величинами, но может работать и с данными категориального типа.

Система автоматически генерирует все гипотезы о зависимостях и связях между параметрами объекта, удовлетворяющие наперед заданным синтаксическим ограничениям. К каждой гипотезе система выдает численные характеристики, по которым можно решить, подходит или не подходит гипотеза для данного случая.

Для управления и использования базы знаний необходимо иметь управляющий блок, главные функции которого можно

определить следующим образом:

- проверка знаний, предложенных для включения в базу знаний, на непротиворечивость и избыточность по отношению к уже хранимым знаниям;
- выбор соответствующих знаний (особенно ввиду предполагаемого существование 3 уровней знаний в базе знаний), необходимых для выдачи ответа на запрос;
- определение на основе логического вывода с помощью выбранных знаний объема данных, которые необходимы извлечь из базы данных.

5. Заключение

Описанная интегрированная система для поддержки исследований в области изучения механизмов учения, должна прежде всего обеспечить комфорт для исследователя при использовании современной вычислительной техники, т.е. обеспечить коммуникацию исследователя посредством непроцедурального языка в диалоговом режиме с базой данных, базой знаний и с проблемно-ориентированными библиотеками программ.

Литература

1. Chvalovský, V.: Expert Systems Design: Pragmatic Approach. Proceedings of Eight. International. Seminar on Database Management Systems, September 30 - October 4, 1985, Piešťany, Czechoslovakia.
2. Fabián, Z., Kufudaki, O., Pelikán, E. etc.: Metody pro vyhodnocování signálů. Závěrečná zpráva za dílčí úkol SPZV III-9-1/3; Report General Computing Centre, Czechoslovak Academy of Sciences, Prague 1985.
3. Kellogg, Ch.: From Data Manegement to Knowledge Management; Computer, Vol.19, January 1986, No.1, pp.75-84.
4. McCarthy, J.: Programs with Common Sense; Mechanization of Thought Processes, Vol.I, London:HMSO 1962.
5. Mittal, S., Chanderasekaranm B., Sticklen, J.: Patrec: A Knowledge-Directed Database for a Diagnostic Expert System; Computer, Vol.17, September 1984, No.9, pp.51-58.
6. Murphy, T.E.: "Setting up Expert System Development Tools". I&SCS Magazine, March 1985, Special section on Artificial Intelligence in Process Control.
7. Vítková, G.: K problematice využití databázových systémů v AVV. Report V-200, General Computing Centre, Czechoslovak Academy of Sciences, Prague 1985.
8. Turing, A.M.: Computing Machinery and Inteligence, Mind, Vol.59, 1950, pp.433-460.
9. Udagawa, Y. and Ohsuga, S.: Developing a Deductive Relational Database for Uniform Handling of Complex Queries. Journal of Information Processing, Vol.6, Nov.1983, No.3, pp.127-137.

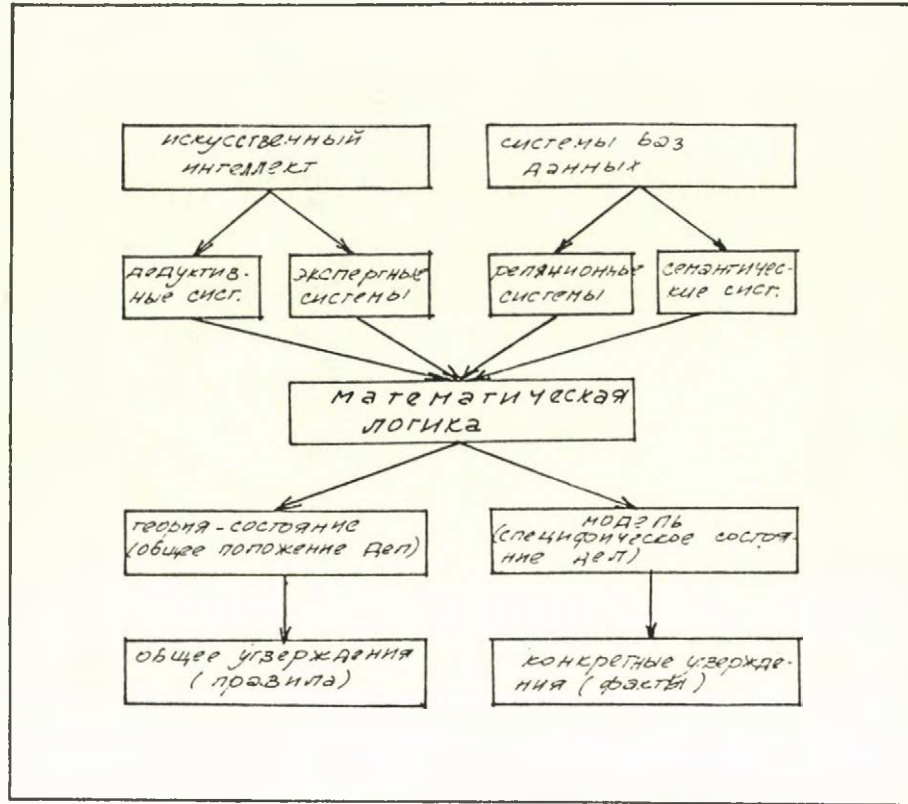


рис. 1

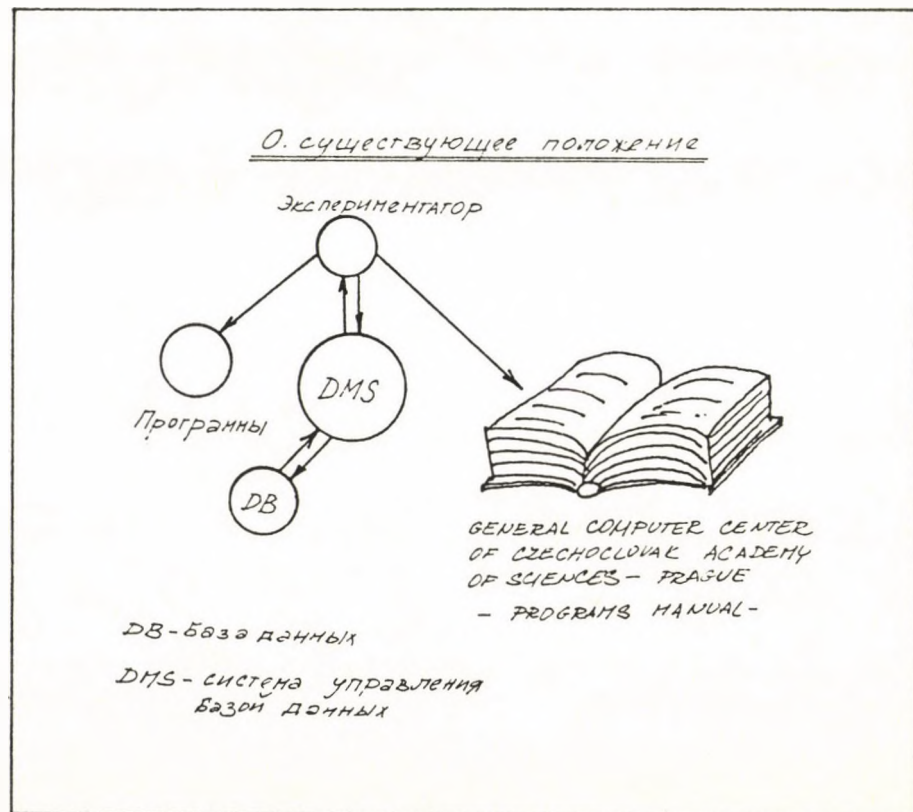
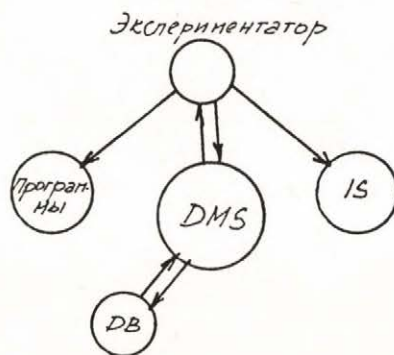
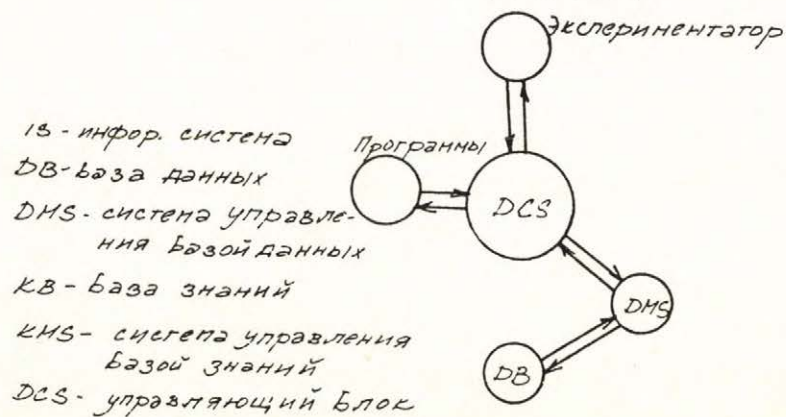


рисунок 2

а) I этап



б) II этап



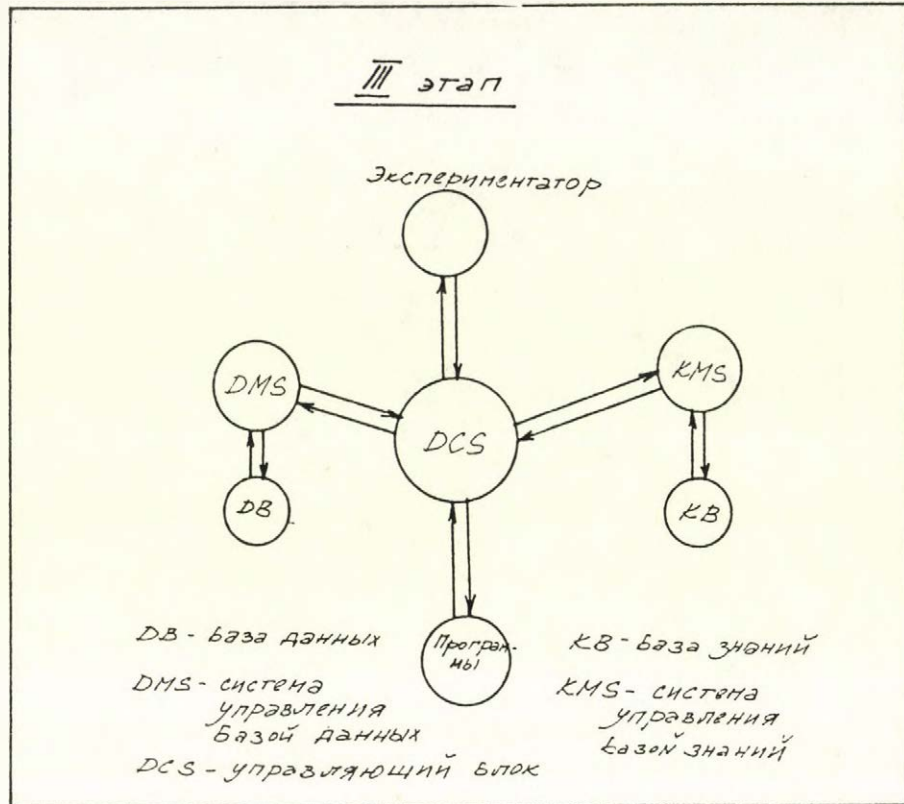


рис. 4.

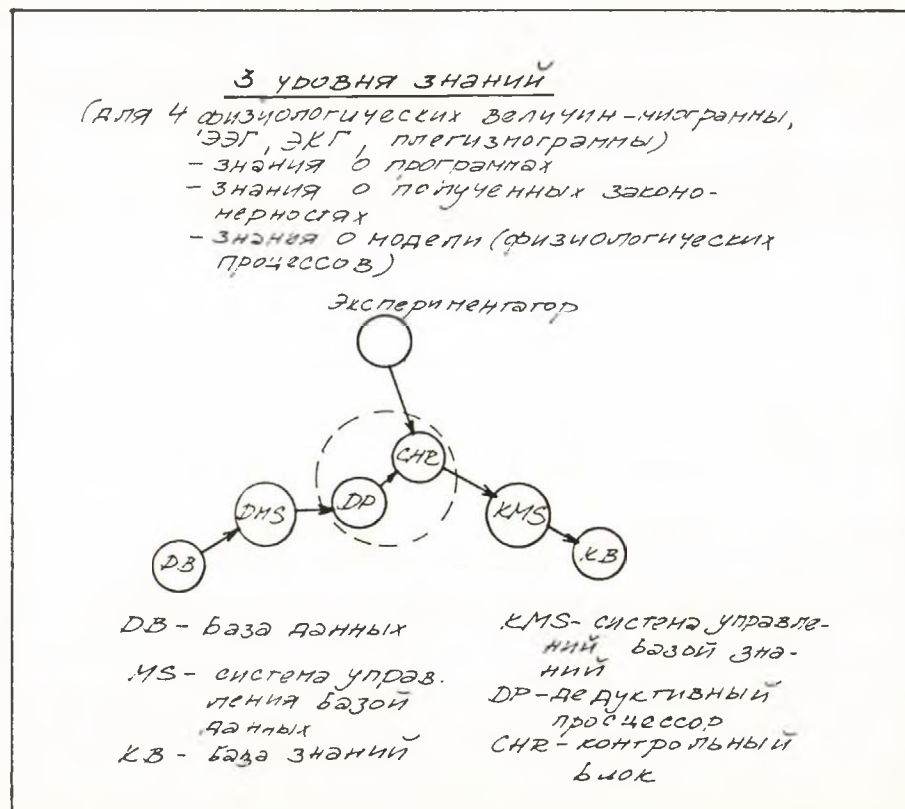


рис. 5

ИСПОЛЬЗОВАНИЕ СЕТЕЙ ЭВМ ПРИ ПРОЕКТИРОВАНИИ
АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ГО-
РОДСКИМ РАЙОНОМ

И.И.Гейци, Г.И.Карпачев

Социальная инфраструктура и производство как взаимодействующие факторы, определяющие социально-экономическое развитие, должны управляться отраслевыми и территориальными органами на согласованной основе. В правильном сочетании отраслевой и территориальной систем управления заложена также возможность эффективного управления межотраслевыми процессами, имеющими огромное экономическое значение. Вместе с тем, на практике, инфраструктура, определяющая социальные условия жизни населения, управляется только территориальными органами, а производство, создающее материальные условия жизни людей, управляется, по-существу, автономно отраслевыми органами. Территориальное управление призвано обеспечить развитие социально-экономического комплекса на данной территории и сбалансированность между ее производственно-экономическими, инфраструктурными и экологическими компонентами. При этом должны быть обеспечены благоприятные условия для воспроизводства трудовых ресурсов, а также гармоничное функционирование всей системы материального и культурного жизнеобеспечения людей, рациональное использование и охрана природных ресурсов. Успешное решение этих задач требует выработки сбалансированных решений по всем упомянутым аспектам деятельности отраслевых и территориальных органов управления /1/. Теоретическая и практическая проработка этих проблем еще далека от завершения /2, 3/. Сложность упомянутой проблемы усугубляется еще и тем обстоятельством, что в территориальных органах управления недостаточно развиты современные методы и средства подготовки информации,

необходимой при согласовании территориальных и отраслевых программ развития.

В территориальные органы управления необходимая для принятия решений информация поступает с меньшей оперативностью и точностью, чем в министерствах и ведомствах, где за последние десятилетия создана достаточно развитая информационно-вычислительная инфраструктура /1 ÷ 3/. Основная причина такого положения обусловлена явно недостаточной "фондовооруженностью" и, как следствие, низкой производительностью труда сотрудников аппарата регионального управления /1, 4/.

Проблема повышения эффективности работы территориальных органов управления должна решаться, в основном, за счет:

- дальнейшего совершенствования организационно-правового механизма, регулирующего взаимоотношения отраслей и территориальных социально-экономических комплексов;

- повышения производительности труда управленческого персонала, а также качества принимаемых решений за счет создания и внедрения современной информационно-вычислительной технологии и применения количественных методов анализа информации /4, 5/. Принимая во внимание специфику территории как объекта управления /источники и потребители информации распределены на значительном пространстве/, создание современной информационной технологии возможно только с использованием территориально-распределенных информационно-вычислительных сетей ЭВМ /6 ÷ 8/.

В докладе обсуждаются проблемы разработки автоматизированной системы управления городским районом на базе территориально-распределенной неоднородной сети ЭВМ коллективного пользования.

Информационные характеристики
городского района

На современном этапе особую актуальность приобретает проблема достоверности информации, обрабатываемой в государственном управлении. Как показали исследования, около 80% социально-экономической информации, используемой на всех уровнях государственного управления перерабатывается в ее первичном звене - административном районе /рис. 1/. В этом контексте особое внимание должно быть уделено работам по созданию автоматизированных систем подготовки оперативной и достоверной информации для районных органов управления /9 + 11/.

Городской район представляет собой сложную социально-экономическую систему, элементы которой тесно связаны большим числом хозяйственных, политических, социальных и других отношений и интересов. Как открытая система он функционирует и развивается в соответствии с его участием в общественном разделении труда. Наиболее тесные внешние социальные и экономические связи городской район имеет с вышестоящим по рангу территориальным звеном - городом, в состав которого он непосредственно входит. Эти связи проявляются через взаимодействие с соседними районами или общегородскими образованиями /1, 11/.

Сложность и многоотраслевой характер городского района определяет многообразие видов информации, используемой в управлении его социально-экономическим развитием. Можно выделить следующие укрупненные группы источников и потребителей информации:

- вышестоящие партийные, советские, общественные и хозяйственные органы;

- предприятия, организации и учреждения района;
- население района.

Анализ существующего документооборота городского района показал наличие избыточных потоков документов между различными органами управления и, вместе с тем, значительный дефицит информации, необходимой для выработки и принятия своевременных управленческих решений /12/. Основные акты управления сводятся к разработке управленческих документов, т.е. "носителей информации, содержанию и способам фиксации которой действующее законодательство придает определенное правовое значение" /13/.

В Советском районе г.Новосибирска с численностью населения 100 тысяч человек действуют около 500 народнохозяйственных объектов 15 министерств и ведомств. В районе используется более 200 различных форм документов, которые относятся к плановой, отчетно-статистической, первично-учетной, финансовой, бухгалтерской, организационно-распорядительной системам документации. Годовой документооборот превышает 100 тыс. документов /без учета внутреннего документооборота предприятий/, около трети которой приходится на население района. Работа с документами занимает в среднем от 30 до 80% рабочего времени руководителей и подавляющую часть времени младшего персонала. Общий объем информации, циркулирующей в районе, колеблется в пределах 4 - 6 млрд. знаков в год /объем хранимой информации составляет 2 - 3 млрд. знаков/ /10, 14/.

Обмен информацией между населением, объектами народного хозяйства района и органами управления происходит, в основном, посредством бумажных документов. Частое переписывание одной и той же информации /например, паспортных данных о человеке, данных о его жилищных условиях, здоровье и т.п./, используемой для различных

адресатов, приводит к снижению ее достоверности и искажению. Время доставки документов курьером или почтой исчисляется сутками. В органах накапливается большой архив документов. Поиск нужного документа, а тем более его оформление, требующее последовательной выборки информации их архива, требует больших трудозатрат и недопустимо затягивает оформление документов.

В связи с этим совершенствование управления городским районом в значительной степени сводится к решению задачи автоматизации сбора, хранения и обработки информации и, в конечном счете, оптимизации внутрирайонного документооборота в целом. В свою очередь, для этого необходимо создать соответствующие технические и программные средства, позволяющие применять количественные методы обработки информации, широко использовать математические модели для анализа возможных ситуаций и выбора наилучших решений.

Использование современной информационной технологии на базе сетей ЭВМ позволяет в несколько раз уменьшить документооборот и трудоемкость подготовки документов при одновременном повышении достоверности информации /4, 7, 12/.

Вычислительная сеть коллективного пользования -
база для создания территориальных автоматизиро-
ванных информационных систем

Как было указано выше, около 80% информации, связанной с государственным /территориальным/ управлением порождается и перерабатывается в районе /рис. 1/. В этой связи чрезвычайно актуальной задачей является создание современных автоматизированных систем районного уровня, позволяющих обеспечить одноразовый ввод первич-

ной информации органу-источнику непосредственно с рабочих мест и её многократное использование всеми заинтересованными абонентами /4/. Эксплуатируемые в настоящее время территориальные АСУ, как правило, эту проблему не решают /5/. По этой причине часть информации вообще не обрабатывается, а другая часть сохраняется или передается в вышестоящие органы управления в усредненном виде, не учитывающем в полной мере специфику условий, сложившихся в районе. Эффективное решение этой задачи возможно только на базе территориально-распределенной вычислительной системы коллективного пользования, путем создания распределенных баз данных и автоматизированных рабочих мест делопроизводства в органах управления районом. При этом создаются условия по выполнению профессиональных обязанностей работниками органов управления на качественно новом уровне, путем коллективного использования информационно-вычислительных мощностей в диалоговом режиме, повышается персональная ответственность сотрудников за достоверность вводимых в ЭВМ первичных данных /4, 11/.

Наши основные разработки направлены на создание автоматизированных рабочих мест для управленческого персонала на базе локальных сетей мини- и персональных ЭВМ, с выходом на региональные информационно-вычислительные сети ЭВМ, системы распределенных банков социально-экономической информации /6, 11, 12/. Это дает возможность решать не отдельные задачи, а комплекс взаимосвязанных управленческих задач с единой информационной базой. Широкое использование подобных систем управленческим персоналом возможно только при существенном облегчении взаимодействия человека с ЭВМ. Режим диалога управленца с ЭВМ должен иметь естественно-языковую основу. Диалоговый режим на языке конечного пользователя приводит

к исключению посредников между ЭВМ и конечным пользователем и позволяет последнему проводить более глубокий анализ ситуаций, возникающих в управленческой практике.

С целью объединения множества ЭВМ в единый комплекс, организации развитой системы дистанционного доступа пользователей к ЭВМ, а также внедрения новых технологических и сервисных средств обработки информации на ЭВМ Вычислительным центром СО АН СССР совместно с СКБ НП и СКБ ВТ СО АН СССР, НФ ИТМ и ВТ и ГПВЦ СО АН СССР создана и передана в эксплуатацию территориально-распределенная неоднородная вычислительная сеть ЭВМ коллективного пользования Новосибирского научного центра СО АН СССР /6/.

При проектировании системы были реализованы следующие основные архитектурные решения: объединение в единый комплекс различных по типу и классу ЭВМ; территориальная распределенность /сетевая структура/ технических средств; переносимость и преемственность прикладного программного обеспечения при развитии системы; ориентация системы на широкий класс, в том числе "непрофессиональных", пользователей.

Технические средства 1 очереди вычислительной сети коллективного пользования /ВСКП/ включают три основных компонента: базовые вычислительные комплексы /БВК/, периферийные центры обработки /ПЦО/ и систему передачи данных /рис. 2/. Основные информационно-вычислительные мощности сосредоточиваются на базовых вычислительных комплексах /ЭВМ БЭСМ-6, ЕС ЭВМ старших моделей/.

Функционально БВК предназначены для производства вычислительных работ, накопления больших массивов информации, поиска информации и дистанционной пакетной обработки по заданиям с ПЦО, формируемым связным процессором.

Многомашинный вычислительный комплекс общего назначения выступает для пользователей как единая информационно-вычислительная система. Комплекс технических средств и общесистемного программного обеспечения обеспечивает пользователям режим разделения времени при контакте с системой и режим пакетной обработки для решения больших задач, некритичных ко времени исполнения. Предусмотрен доступ к отдельным ЭВМ, входящим в БВК и их штатным программным средствам как удаленным, так и локальным пользователям с помощью собственной терминальной сети базовой ЭВМ.

Периферийные центры обработки созданы на базе СМ ЭВМ и функционально обеспечивают: прием, предварительную обработку и преобразование к каноническому для системы виду поступающей от абонента информации и управляющих директив; первичную диспетчеризацию потоков заданий и сбор находящейся в его компетенции статистической информации для ВСКП в целом; прием адресуемой абоненту информации и ее выдачу в требуемом формате /6/. ПЦО имеют собственную локальную абонентскую сеть, концентрирующую пользователей по соответствующей предметной области.

Общесистемное программное обеспечение ВСКП представляет собой комплексы программных средств, обеспечивающих функционирование как отдельных компонент /БВК и ПЦО/, так и ВСКП в целом. Программное обеспечение БВК ЕС включает операционную систему ОС ЕС версий 4.1 и выше, систему планирования КРОС и пакеты прикладных программ, расширяющие возможности ОС ЕС в части описания и манипулирования сложноструктурированными данными. Программы, обеспечивающие взаимодействие ОС ЕС ЭВМ, входящих в состав БВК, со связным процессором БВК полностью базируется на штатных возможностях ОС ЕС.

Программное обеспечение СМ ЭВМ включает операционную систему ОС РВ и системный комплекс NIKE, предназначенный для абонентов ВСКП, работающих за терминалами ПЦО, и позволяет /11/: формировать задания ЕС ЭВМ БВК на языке управления работами; хранить, редактировать и распечатывать задания; передавать задания на ЕС ЭВМ БВК и формировать входную очередь заданий; получать информацию о выполнении задания; передавать информацию от ЕС ЭВМ БВК на терминалы пользователей.

Созданные программно-технические средства в совокупности обеспечивают отработку основных элементов типовой автоматизированной информационной системы городским районом как по составу и содержанию задач, так и по технологии их реализации.

Информационное и технологическое обеспечение АСУ-район

Информационную основу АСУ-район составляют первичные данные о его жителях, предприятиях и организациях.

Решение задачи информационного обеспечения АСУ-район мы видим на пути формирования общерайонной базы данных коллективного пользования. Поэтапное создание и дополнение общерайонной базы данных локальными базами, принадлежащими отдельным органам управления, позволит в перспективе решать многие задачи управления с одной стороны минимизируя общие трудозатраты, и с другой обеспечивая необходимую степень автономности подсистем.

При проектировании современной технологии сбора, хранения и обработки данных в городском районе нами ставилась задача создать такой автоматизированный инструмент, который бы позволил работникам территориального управления /8/:

- осуществлять общение с информационной базой в режиме диалога;
- обеспечивать автоматизированное ведение делопроизводства, прежде всего документов, являющихся источниками накапливаемой информации, которая многократно используется разными подразделениями
- минимизировать эксплуатационный персонал АСУ-район.

На базе терминальной сети мини-ЭВМ СМ-4, входящей в состав ПЦО райисполкома /рис. 3/, созданы диалоговые системы, позволяющие осуществлять одноразовый ввод информации непосредственно с рабочего места с первичных документов, исключая ручную подготовку к вводу в ЭВМ /8, 12/.

Одной из наиболее сложных задач, в которой использована обсуждаемая информационная технология, было создание банка паспортных данных о жителях района. Автоматизированный банк данных "Население" эксплуатируется на средствах сети ЭВМ и использует современные языки описания и манипулирования данными на ЕС ЭВМ /СУБД ИНЕС/ /8, 12/.

Банк данных о населении во взаимосвязи с другими подсистемами представляет собой межотраслевую многоцелевую информационную систему, предназначенную для решения ряда актуальных задач управления экономикой района, в том числе:

- планирования перспектив развития производственной сферы на основе прогноза его обеспеченности трудовыми ресурсами;
- комплексного развития социальной инфраструктуры района с непрерывным учетом потребностей населения в различных видах культурного и бытового обслуживания и др.;
- сокращения времени приема на работу, выдачи различных справок, переписи населения и т.п. за счет организации соответствующей автоматизированной информационной службы;

- улучшение медицинского обслуживания всех категорий населения, в частности, за счет всеобщей диспансеризации, улучшения профилактического и поликлинического обслуживания.

В технологическом отношении подсистемы АСУ-район можно разбить на две группы /рис. 4/. Первая группа содержит относительно автономные подсистемы, которые имеют сравнительно слабые информационные взаимосвязи, а суммарный объем хранимой информации не превышает 10÷20 Мбайт /"Контроль", "Капстроительство" и др./. Эти подсистемы реализованы полностью на мини-ЭВМ и доступны руководству в диалоговом режиме. Вторую группу составляют подсистемы, использующие общие массивы информации /"Плановые расчеты", "Население", "Жилищно-коммунальное хозяйство", "Паспорт района", "Паспорта предприятий" и т.д./. Реализация подсистем этой группы потребовала больших вычислительных мощностей /ЕС ЭВМ 1052/ и хранения значительных /несколько сот Мбайт/ информации.

Применение новой технологии обработки информации позволяет улучшить качество решения задач территориального управления за счет повышения уровня комплексности в развитии городского района, оптимизации управленческих решений, оперативности информационного обслуживания, роста производительности труда управленческих работников и других факторов.

Проблемы эксплуатации и перспективы развития АСУ-район

Наибольший эффект от внедрения АСУ-район следует ожидать от создания общерайонного информационного фонда, содержащего данные из реально существующего в районе документооборота /8/. В решении этой проблемы должны участвовать практически все предприятия и уч-

реждения района. Однако эта задача сильно затруднена межведомственными барьерами, а также тем, что ряд наиболее значимых предприятий района не выполняет известных постановлений директивных органов о предоставлении местным органам информации.

Серьезной проблемой в организации эксплуатации является наличие у пользователей психологического барьера, обусловленного их недоверием к возможностям информационной технологии и недостаточным пониманием своей роли в этой системе. В связи с этим, важным требованием, предъявляемым к программно-техническим средствам и ко всем задачам АСУ-район является удобство пользования и легкость их освоения.

Так как источники и потребители информации распределены по всей территории района, то повышенные требования предъявляются к надежности функционирования сети ЭВМ, необходимо отработать комплекс мер по защите информации и обеспечению ее безопасности.

В текущей пятилетке перед нами ставится задача оснащения работников органов управления автоматизированными рабочими местами на базе терминальной сети мини- и микро-ЭВМ, позволяющими им выполнять свои профессиональные обязанности в режиме непосредственного общения с ЭВМ на языке, имеющем естественную основу, без участия обслуживающего персонала. Общерайонный информационный фонд должен стать базой не только для обеспечения оперативного управления, но и основой создания комплекса моделей по перспективному планированию и проигрыванию различных вариантов социально-экономического развития района.

Наши усилия по дальнейшему развитию программно-технических средств АСУ-район направлены на создание автоматизированных рабочих мест на базе персональных ЭВМ, объединяемых в локальную сеть

микро- и мини-ЭВМ /рис. 3/ /15/. Заканчивается разработка системы автоматизации делопроизводства /микро-ЭВМ типа "Электроника-60"/, предоставляющей удобный интерфейс для конечного пользователя с простым набором команд на русском языке, экранным редактированием команд и данных, с возможностью генерации стандартных выходных форм.

Заключение

На базе неоднородной сети ЭВМ создана региональная информационно-вычислительная система, позволяющая осуществлять распределенную обработку информации непосредственно с рабочих мест конечных пользователей различных предметных областей. Данная система была использована при создании первой очереди АСУ Советским районом г.Новосибирска.

Проведено всестороннее изучение документооборота и информационных потоков городского района. Спроектированы и сданы в опытную эксплуатацию подсистемы первой очереди АСУ-район. Проектные решения апробировались в практической работе аппарата райисполкома при решении задач, связанных с оперативным управлением и перспективным планированием социально-экономического развития района. Передано в опытную эксплуатацию около 150 задач, охватывающих общерайонный и отраслевые уровни управления.

Исследованы возможности архитектурных, программных и технических решений для повышения достоверности и оперативности информации, создания интегрированных баз данных общего назначения.

Полученные в ходе работ практические результаты и накопленный опыт создали реальные предпосылки для проектирования типовой автоматизированной информационной системы городского района сетевого типа, как базового элемента для информационных систем более высоких уровней /город, область/.

ЛИТЕРАТУРА

1. Иконников В., Крылов С. О сочетании отраслевого и территориального управления. Коммунист. М., №4, 1984.
2. Оприско Ю.И. Системотехнические проблемы координированного управления городом.- В сб. Тезисы докладов II Всесоюзной конференции "Управление большим городом". М., 1983, ч.1, с.17-20.
3. Черкасов Ю.М., Поздняков В.В. Отставание координации от специализации - особенность развития больших организационных систем.- М.: Сборник научных трудов НПО АСУ "Москва", 1981.
4. Ершов А.П. Автоматизация работы служащих. Микропроцессорные средства и системы. 1985, №2, с.6-15.
5. Перегудов Ф.И., Тарасенко В.П. и др. Автоматизированная система управления хозяйством Томской области /опыт разработки и внедрения/. М., Управляющие системы и машины. 1986, №1, с.113-117.
6. Марчук Г.И. и др. Территориально-распределенный многомашинный вычислительный центр коллективного пользования СО АН СССР: Препринт №245,- Новосибирск, 1980,- 59 с.- В надзаг: ВЦ СО АН СССР.
7. Лорин Г. Распределенные вычислительные системы. М.: "Радио и связь", 1984.
8. Глазырин М.В., Гейци И.И. и др. Опыт и перспективы создания АСУ Советским районом г.Новосибирска.- В сб.: Тезисы докладов II Всесоюзной конференции "Управление большим городом". М., 1983, ч. II, с. 64-66.
9. Громов Г.Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации.- М.: Наука, 1984.

10. Братухин П.И. и др. Основы построения больших информационно-вычислительных сетей.- М.: Статистика, 1976.
11. Алексеев А.С., Гейци И.И., Карпачев Г.И. и др. Автоматизация управления городским районом: Препринт №621. - Новосибирск, 1985.- 27 с. В надзаг.: Вычислительный центр СО АН СССР.
12. Карпачев Г.И. Распределенные информационные системы в территориальных АСУ: Автореферат диссертации на соискание ученой степени кандидата технических наук /01.01.10/.- Новосибирск, 1984,- 24 с.
13. Венгеров А.Б. Право и информация в условиях автоматизации управления /теор. вопр./.- М.: Юрид. лит., 1978.- 208 с.
14. Цицин П.Т. Организация управления экономикой городского района.- М.: Экономика, 1978.
15. Елинер Э.И., Клименко А.Д., Костылев Д.А. Локальная сеть ВЦКП СМ-2 на базе сетевых интерфейсов DL-NET и сетевого матобеспечения АЛИСА /Концепция/. М.: Микропрцессорные средства и системы, 1986 /в печати/.

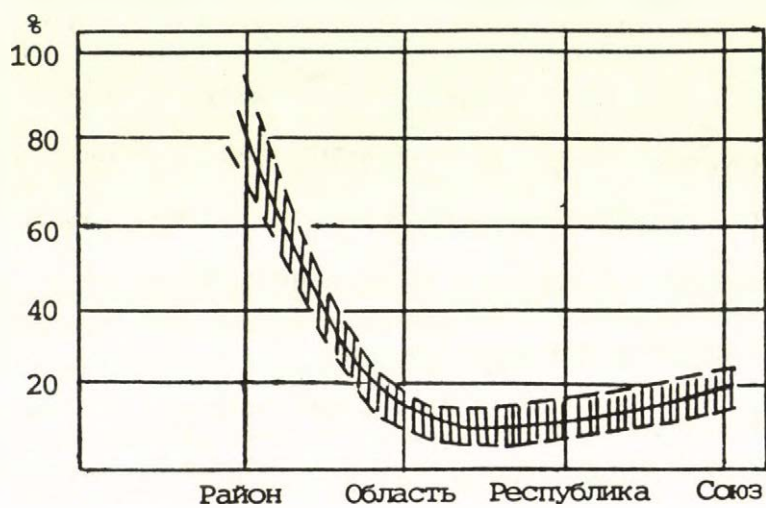


Рис. 1 Распределение объема входной информации по всем задачам, решаемым на каждом из уровней государственного управления

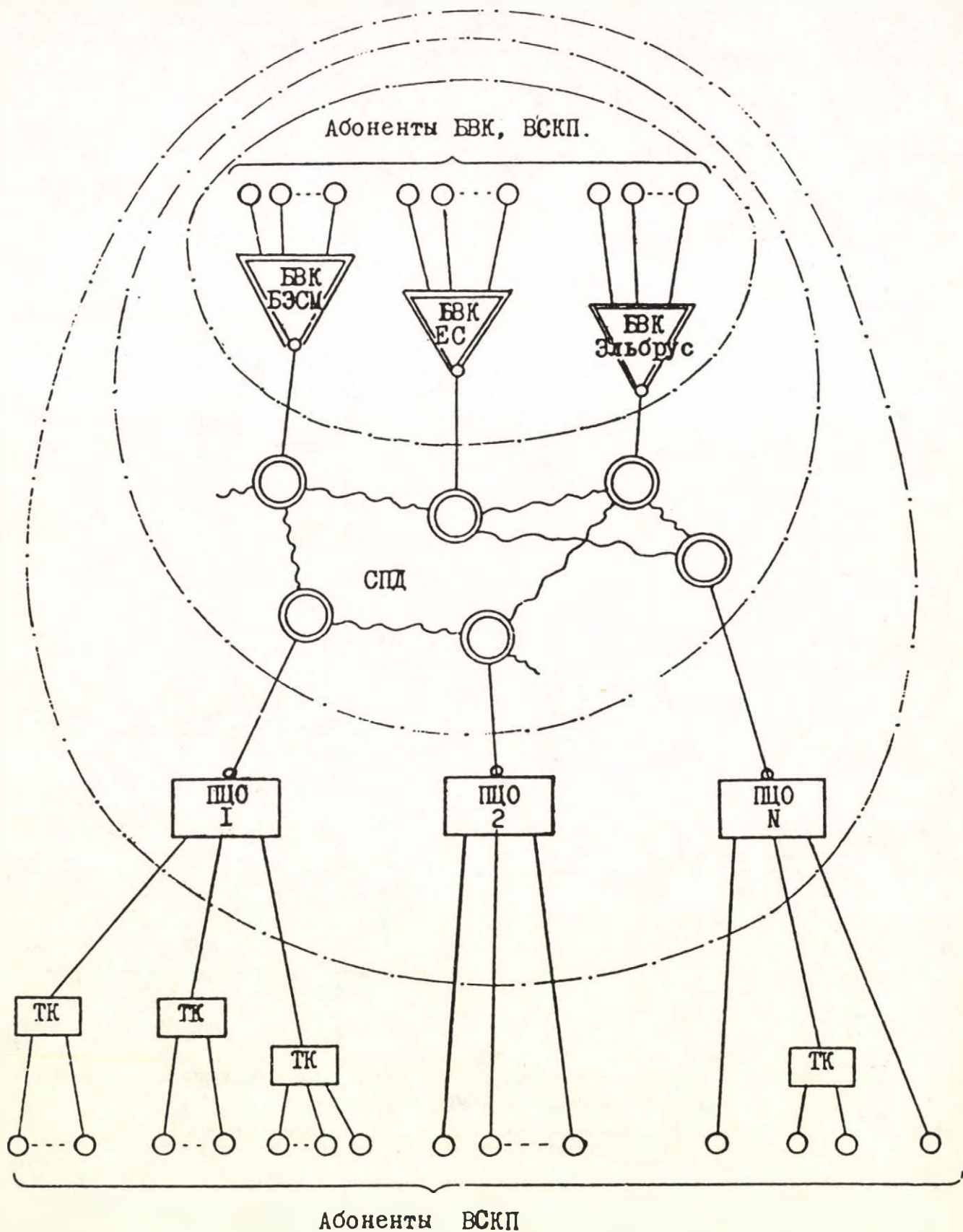


Рис. 2 Функциональная схема первой очереди ВСКП Новосибирского научного центра СО АН СССР

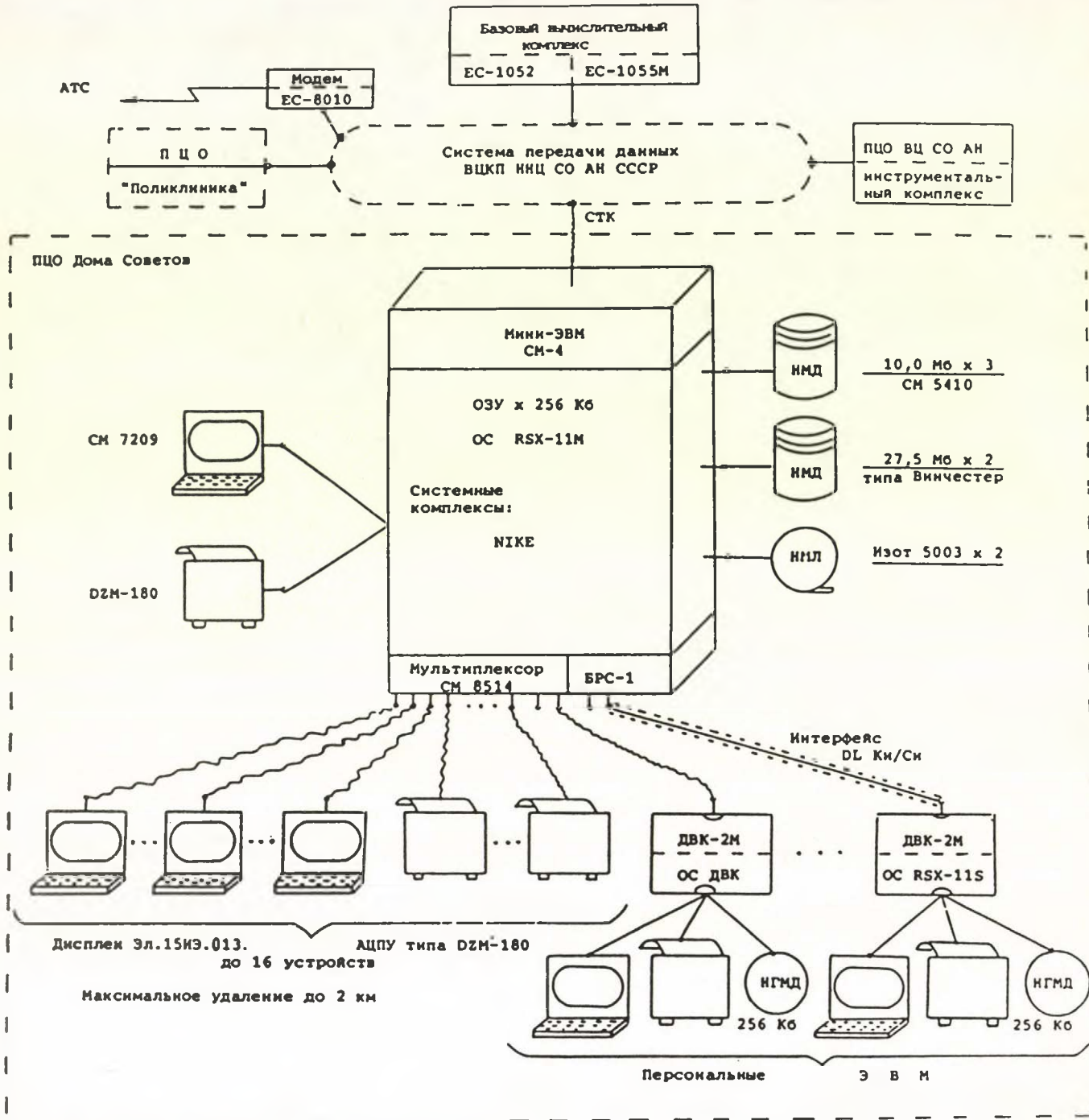


Рис. 3 Структура технических средств информационно-вычислительной сети АСУ Советским районом г.Новосибирска

/ПЦО - периферийный центр обработки; СТК - сетевой контроллер;
АТС - автоматическая телефонная станция; БРС - блок расширения системы/

ИМИТАЦИОННАЯ МОДЕЛЬ ДЛЯ ПРОВЕРКИ РЕАЛИЗУЕМОСТИ ПЛАНА ПОСТАВОК ПРИ УПРАВЛЕНИИ ПРОДУКТОПРОВОДОМ ДЛЯ ТРАНСПОРТИРОВКИ НЕСКОЛЬКИХ НЕФТЕПРОДУКТОВ

Инзелт, П. - Таполцаи, Л.

Исследовательский институт вычислительной тех-
ники и автоматизации ВАН, Будапешт

I. Введение

В сообщении [1] дано достаточно детальное описание проблемы управления продуктопроводом для транспортировки нескольких нефтепродуктов. Обоснованы практическое значение и важность решения этой проблемы; поставка максимального количества жидких нефтепродуктов по продуктопроводу может дать очень значительный экономический эффект.

В сообщении [1] показывается, что различные нефтепродукты могут поставляться последовательно, "пробками"; при этом необходимо совместно решать задачу дискретного характера (определение последовательности пробок) с задачей непрерывного характера (определение длины, объема этих пробок) при соблюдении набора технологических ограничений. Целевой функцией может являться минимум стоимости транспортировки нефтепродуктов в плановом периоде по всему региону, что приблизительно эквивалентно максимальному удовлетворению спроса по продуктопроводу по всем маркам. В результате того, что еще не найден эффективный алгоритм для решения задачи данного класса, одним из реальных путей ее решения выбирали интерактивный способ, основанный на составлении плана поставки диспетчером с использованием его опыта и рассчитанной информации (то есть исходная последовательность пробок задается диспетчером), получении реализуемого плана поставок методом имитационного моделирования и локальной оптимизации этого плана линейным программированием. Поставленной целью являлась разработка такого программного средства, которое даст эффективную помощь диспетчеру продуктопровода.

2. Исходные условия

Здесь не будем дать полный перечень условий работы продуктопровода, отметим только, что действительны те основные технологические условия эксплуатации продуктопровода, которые описаны в сообщении [1], кроме того, перечисляются исходные условия, имеющие принципиальное значение с точки зрения решения задачи.

2.1. Предполагается, что задача управления продуктопроводом решается в рамках иерархической системы принятия решения, следовательно, задан плановый период и план на поставку продуктов по всем пунктам потребления (ПП) в плановом периоде. Ввиду того, что пункты потребления играют роль достаточно большого буфера по сравнению с неравномерностью отгрузки продуктов, можно предполагать, что в данном плановом периоде (но только в этом периоде) отгрузка продуктов с пунктов потребления идет с постоянной, заданной интенсивностью.

2.2. Один из продуктов – в нашем случае газойл – поставляется по продуктопроводу в количестве, значительно превосходящем количество других продуктов (автомобильных бензинов различных марок и бензина для пиролиза). Нормальный цикл работы продуктопровода выглядит следующим образом:

- а). Продуктопровод наполнен газойлом,
- б). Поставляются бензины к пунктам потребления "пробками",
- в). Продуктопровод снова наполняется газойлом и транспортировка может быть приостановлена.

2.3. По продуктопроводу сохраняется постоянное давление, линейная скорость продукта зависит только от сечения трубы (труб), по которой (которым) в данный момент отбирается продукт. Скорость отбора является заданной константой.

- 2.4. Допускается одновременный отбор не более в двух (трех) пунктах потребления.
- 2.5. Все технологические характеристики и ограничения системы известны (длины участков трубопровода, диаметры участков, емкости резервуаров, запасы в начальном моменте поставки и т.д.).

Можно исходить из того, что полученный план на поставку продуктов реализуем. Действительно, речь идет о работающей системе, можно рассчитать коэффициент нагрузки продуктопровода [1] и есть опыт по реальному его диапазону. Некоторые тривиальные условия в начале проверяются программой (этой проверкой не будем заниматься) и диспетчер также задает реальную последовательность пробок. Естественно, задача может не иметь решение при этой последовательности пробок, но есть основание предполагать – и это подтверждалось опытом – что вторая-четвертая "попытка" диспетчера обязательно приводит к субоптимальному решению или имитационное моделирование явно показывает то узкое место, которое должно устраняться. Несомненно, имеется теоретическая возможность того, что описанная интерактивная процедура не приводит к решению, но в практике это достаточно маловероятно.

3. Задача поставки единственного продукта по продуктопроводу

Данная задача по сложности несравнима с задачей управления многопродуктовым продуктопроводом, в то же время небезсмысленно исходить из этой задачи по двум причинам:

- она имеет практическое значение: как отмечалось, трубопровод обычно наполнен газойлом и многопродуктовый этап всегда начинается с "распределением" газойла, опорожнением трубопровода;
- в части решения задачи многопродуктового продуктопровода данная задача является исходной, облегчает понимание более сложной задачи.

По продуктопроводу поставляется только газойл, задан плановый период $T = \{t_0, t_v\}$ и вектор столбец $B = \{b_1, b_2, \dots, b_n\}$, характеризующий план поставок по пунктам потребления (их всего n). Ввиду того, что нельзя одновременно отбирать газойл по всем пунктам потребления (см. п. 2.4), поставка будет реализоваться в нескольких этапах. Этапом называется определенный режим отбора газойла из продуктопровода, причем каждому этапу соответствует какой-то промежуток времени.

Можно записать следующую систему линейных уравнений:

$$\underline{V} \cdot \underline{X} = \underline{B} \quad (I)$$

где: V - $n \cdot s$ - мерная матрица режимов отбора, $V = \{v_{ik}\}$, $\text{м}^3/\text{час}$

X - вектор промежутков времени, $X = \{x_1, x_2, \dots, x_s\}$, час

B - вектор столбец плана поставок по пунктам потребления в плановом периоде, $B = \{b_1, b_2, \dots, b_n\}$, м^3

i - порядковый номер пункта потребления,

k - порядковый номер промежутка времени.

Матрица \underline{V} задается диспетчером, ее элементы представляют в себе скорость отбора продукта $\text{м}^3/\text{час}$ по пунктам потребления. В каждом столбце не более двух элементов могут отличаться от 0 (см. п. 2.4), сумма элементов по столбцам не должна превосходить максимальную пропускную способность продуктопровода w_{cs} , далее величина конкретного элемента не должна превосходить максимальную скорость отбора w_i на данном пункте потребления. Кроме этого, естественно, в каждой строке должен быть хотя бы один элемент, отличающийся от 0.

На основе данной системы линейных уравнений можно определить промежутки времени \underline{X} , соответствующие конкретным режимам поставки, причем последовательность этих режимов никакую роль не играет.

По другому обстоит дело, если вводятся простейшие ограничения, характеризующие реальную ситуацию. В рамках планового периода в каждом моменте требуется следить за тем, чтобы резервуары в

пунктах потребления не пустовали и не переливались. Так как наполнение резервуаров идет с постоянной скоростью и отгрузка также осуществляется с постоянной скоростью, эти условия достаточно проверить в конце каждого этапа:

$$0 \leq q_{i0} + \sum_{f=1}^k (v_{ij} - z_i) \cdot x_f \leq M_i \quad (2)$$

$i=1, 2, \dots, n$
 $k=1, 2, \dots, s$

где: q_{i0} - запасы в каждом пункте потребления в момент времени t_0 , m^3

M_i - емкость резервуаров в пунктах потребления, m^3

z_i - отгрузка с пунктов потребления, $m^3/\text{час}$

Данная система $2 \cdot n \cdot s$ ограничений будет иметь вид набора прямоугольных диагональных матриц.

Для полноты запишем дополнительные ограничения:

$$\sum_{k=1}^s x_k \leq T \quad (3)$$

$$x_k \geq \epsilon, \quad k=1, 2, \dots, s \quad (4)$$

Целевой функцией может быть минимум времени выполнения плана:

$$F = \sum_{k=1}^s x_k \rightarrow \min \quad (5)$$

Задача решается методом линейного программирования, но обратим внимание, что оно может не иметь решение! Дело в том, что решаемость задачи зависит от последовательности режимов, то есть от очередности столбцов матрицы, кроме того, для решаемости может быть необходимо пользоваться одним и тем же режимом в несколько раз! Как было сказано, матрица режимов v заполняется диспетчером, но от него можно только ожидать реальные режимы отбора, но не полный набор возможных режимов или правильную их последовательность.

В данной ситуации необходимо увеличивать размеры задачи, преобразуя матрицу V в матрицу V^* путем повторения столбцов исходной матрицы, это обеспечивает возможность "возвращения" к раннее использованному режиму. Опыт показывает, что достаточно повторять матрицу в 2-3 раза: действительно, маловероятно, что к одному и тому же режиму необходимо возвращаться больше трех раз. Можно делать и другие преобразования, записать в матрицу периодически нулевые столбцы соответствующие перерыву транспортировки по трубопроводу. Размерность задачи остается в полне допустимых пределах, даже принимая во внимание соответствующее увеличение количества ограничений (2).

Что касается полученного решения, промежутки времени, соответствующие "неиспользованным" режимам, будут равны нулю, то есть получим последовательность режимов (отбора газойла по пунктам потребления) и временные интервалы этих режимов. Решение будет субоптимальным и зависит от набора режимов, заданного диспетчером.

4. Многопродуктовая поставка по продуктопроводу

Многопродуктовая поставка, когда все продукты, за исключением одного (в нашем случае газойла), поставляются "пробками", в отличие от предыдущей задачи, не может быть описана с помощью линейной или линеаризованной модели. Действительно, необходимо вначале определить последовательность пробок, причем эта последовательность зависит и от топологии продуктопровода, то есть в связи с необходимостью опорожнения трубопровода от газойла и удаления "пробки" в полном количестве, будут и "вынужденные" пробки, очередность и объем которых предопределяется сложным образом. Задача управления многопродуктовой поставкой - зная плановые данные по поставке на плановый период - решается нами в следующих этапах:

- определение исходной последовательности пробок и составление исходного плана цикла поставок;

- проверка исходного плана поставок имитационным моделированием, корректировка этого плана в процессе моделирования при необходимости;
- автоматизированное составление модели линейного программирования на основе параметров реализуемого плана, определенного имитацией, с целью улучшения (локальной оптимизации) этого плана.

В дальнейшем более детально описываются эти этапы и ход решения иллюстрируется примерами.

4.1. Составление исходного плана поставок

Данный этап начинается с того, что диспетчер продуктопровода задает плановые данные по поставке продуктов на плановый период и задает исходную последовательность адресованных пробок, то есть задается, что пробка с данным порядковым номером в какие ПП распределяется. Задаются возможные режимы отбора: в каких комбинациях допускается параллельный отбор из трубопровода и какими объемными скоростями.

Для составления последовательности пробок диспетчер получает помощь от системы: так как текущие запасы в системе известны, можно рассчитать те моменты, когда эти запасы кончаются. По газойлу текущие запасы корректируются с содержанием соответствующего участка трубопровода и данный временный ряд можно пересчитать – исходя из средней скорости поставки – в последовательность позднейших моментов отправления данной пробки от пункта снабжения.

Полученная последовательность моментов времени ориентирует диспетчера, но автоматически не определяет последовательность пробок. Например, часто целесообразно исключить сам газойл из многопродуктового цикла: если запас по газойлу кончается во время многопродуктового цикла, целесо-

образно начать этот цикл с газойлом (или, если продуктопровод "стоит", качать туда немедленно газойл). Если какие-то запасы не кончатся в рамках планового периода, то можно этот ПП по данному продукту вообще исключить из цикла, или наоборот - поставлять туда продукт, если это "вписывается" в план поставки.

Отметим, что все данные, которые задаются диспетчером, проверяются на реализуемость. Например, если допускается параллельный отбор, то сумма двух отборных скоростей не должна превосходить допустимую скорость транспортировки по продуктопроводу или для всех ПП и продуктов, которые имеются в плане поставок, должен задаваться хотя бы один режим, когда такой отбор допускается.

Последовательность пробок, заданная диспетчером, преобразуется программой в исходный план поставок. Этот план всегда начинается с переменным "простоем" трубопровода (его величина окончательно определяется во время оптимизации), кроме того включаются в план поставки упомянутые "вынужденные" пробки, определенные топологией системы (опорожнение и наполнение участков), их количество вычитается из планового количества "свободных" пробок. Кроме того, исходя из таблицы допустимых режимов отбора, определяется, какие пробки отбираются параллельно. Для этого используется эвристический алгоритм (например, в начале параллельного отбора одной и той же пробки необходимо быстрее обеспечить поставку к пункту потребления, ближе к пункту снабжения, потому что "ушедшую" пробку вернуть уже нельзя). Отметим, что одной и той же исходной последовательности пробок может соответствовать достаточно большой набор автоматически составленных исходных планов поставок, окончательная селекция которых также является задачей диспетчера. Опыт показывает, что целесообразно сохранять 3-4 исходных плана поставок.

Вышеизложенное проиллюстрируем на простом примере. Пусть имеется продуктопровод, изображенный на рис. 1, причем пункт потребления № 1 настолько близко к продуктопроводу, что объемом соединительного трубопровода (и вообще его существованием) пренебрегаем. Известна таблица допустимых режимов отбора (табл. 1) и задача заключается в том, что необходимо поставлять бензин одной марки в ПП 1, 2 и 3 в количествах b_1 , b_2 , b_3 , соответственно. Последовательность пробок: пункт потребления №№ 1, 2, 3.

При этом один из возможных вариантов исходного плана поставок будет выглядеть следующим образом:

№ операции	Описание операции
1	"Перерыв"
2	Газойл в ПП 1
3	Газойл в ПП 1 и 2
4	Бензин в 1, газойл в 2
5	Бензин в 2, газойл в 3
6	Бензин в 1 и 2
7	Бензин в 2 и 3
8	Бензин в 3

Интересно отметить, что – при заданных скоростях отбора – время операции 4 или 5 строго определено объемом участков В и С. Аналогично определено суммарное время операций 2 и 3 – оно определено объемом участка А. Включение операции 2 увеличивает гибкость плана, если цель заключается в минимизации времени цикла, она будет использоваться только при необходимости. По аналогичному рассуждению можно включить в план после третьей операции операцию "Бензин в 1" и т.д. Изменением последовательности операций 5 и 6 получаем другой план, который при имитационном моделировании будет иметь другие характеристики.

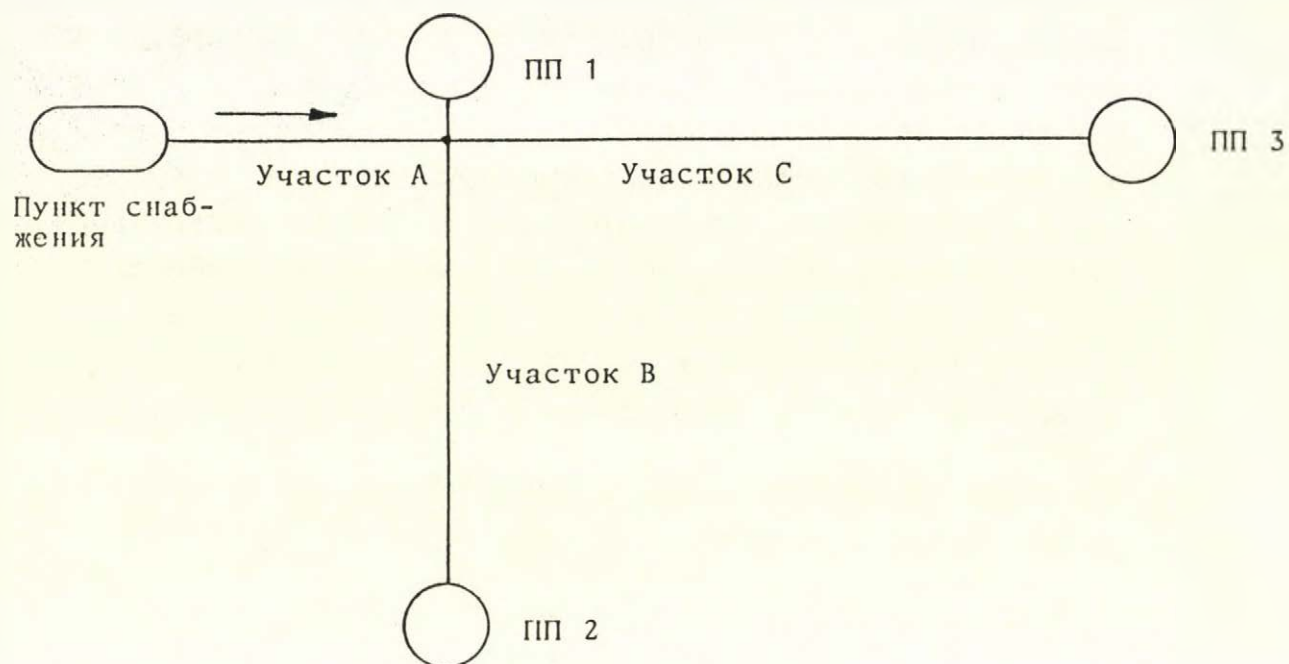


Рис. 1. Принципиальная схема продуктопровода

Таблица 1.

Таблица допустимых режимов отбора, м³/час

№№ режима №№ ПП	1	2	3	4	5	6
1	400	0	0	300	300	0
2	0	500	0	400	0	350
3	0	0	500	0	400	350

Обратим внимание на то, что операции 7 и 8 обеспечивают наполнение системы газойлом, поэтому они необходимы.

Если диспетчером задается последовательность снабжения ПП 3, 2, 1, то получается совершенно другие планы например:

№ операции	Описание операции
1	"Перерыв" ПП
2	Газойл в 2 и 3
3	Газойл в 3 (если участок С больше В), бензин в 2
4	Бензин в 2 и 3
5	Бензин в 1 и 3
6	Бензин в 1 и 2
7	Бензин в 1
8	Бензин в 2 и 3
9	Бензин в 3

Здесь также обращаем внимание на необходимость операций 8 и 9 – они обеспечивают наполнение системы газойлом. Следовательно, в ходе операций 4–6 следует поставлять бензин, соответствующим образом меньше планового количества.

4.2. Имитационное моделирование исходного плана поставок

Имитационное моделирование служит для проверки реализуемости исходного плана поставок. Имитация осуществляется таким образом, что в заданных моментах времени (с небольшим временным шагом, в конкретной реализации он равен 10 минут) проверяется актуальное состояние системы и регистрируются моменты "событий". Событиями считаются следующие явления:

- фронт новой пробки пришел к ответвлению или пункту потребления (здесь событием считается и начальный момент нагнетания нового продукта в пункте снабжения);

- опорожнение резервуара в пункте потребления;
- переполнение резервуара в пункте потребления;
- переход к другой операции (изменение положений заслонок) после отбора планового количества по предыдущей операции;
- пробка пошла мимо пункта потребления и плановое количество данного продукта не отбиралось и т. д.

Имитационная программа прерывает свою работу при обнаружении недопустимой ситуации. Она подготовлена к эвристическому анализу ситуации и модификации исходного плана поставок. Например, опорожнение резервуара возможно по двум причинам:

- время предыдущих операций слишком большое или
- в случае разветвления соединительный трубопровод "блокирован", наполнен другим продуктом.

В первом случае программа уменьшает перерыв, во втором случае - включает в план поставку операции, устраняющую данную "блокировку" и начинает имитацию скорректированного плана. Рассмотрим в качестве примера имитационное моделирование первого варианта описанного исходного плана. Основными событиями будут следующие:

№ события	Описание события
I	Фронт бензина появлялся в пункте снабжения, открыта заслонка к пункту потребления I
2	Открыты заслонки к ПП I и разветвления к ПП 2
3	Фронт бензина появлялся у ПП I (бензин в ПП I, газойл в ПП 2)
4	Фронт бензина появлялся у ПП 2
5	Закрывается заслонка ПП I, открывается заслонка ПП 3 в (бензин в ПП 2, газойл в ПП 3)

№ события	Описание события
6	Открывается заслонка ПП I, закрывается заслонка ПП 3 (бензин в ПП I и ПП 2)
7	Начинается нагнетание газойла с пункта снабжения
8	Закрывается заслонка ПП I, открывается заслонка ПП 3 (бензин в ПП 2 и 3)
9	Фронт газойла дошел к ПП 2, закрывается заслонка к ПП 2)
10	Фронт газойла дошел к ПП 3, останов.

Возможно, конечно и такое событие, что между 3-им и 4-ым событиями обнаруживается окончание запаса бензина в ПП 2. При этом необходимо уменьшить перерыв, а если это невозможно, то уменьшить продолжительность предыдущих операций, например, исключить операцию № 2.

Конечно, сходимость такой процедуры не гарантирована. Может случиться, что при исключении операции № 2 пропорциональная часть (соответствующая операции № 3) содержимого участков А и В переполняет резервуар в ПП 2. При этом включается в план поставок операция "Газойл в 2 и 3", которая по эффективности превышает операцию "Газойл в ПП I".

Наиболее сложной ситуацией является остановление границы фазы, когда возможны четыре различных ситуаций и необходимо искать соответствующее решение. В случае нашего примера предположим, что операция № 4 задана таким образом, что в ПП I следует поставлять только небольшое количество бензина и это время недостаточно для опорожнения разветвления к ПП 2. Операция № 4 кончается, когда поставлен бензин заданного количества, а операцию № 5 нельзя начать, потому что нет бензина у ПП 2, наоборот: в разветвляющем участке имеется граница фаз. Если нельзя увеличивать план поставки бензина в ПП I (например, резервуар переливается), то необходимо перерабатывать план поставок. Такие грубые

ошибки, конечно, обнаруживаются при составлении плана поставок, но аналогичные ситуации нередкие в более сложных задачах.

Из сказанного видно, что имитационное моделирование совершенно свободно манипулирует с планом поставок, оно имеет единственную цель: получить реализуемый план, при котором нет переполнения или опорожнения резервуаров. Полученный результат может отличаться от исходного плана в конкретных операциях, в их последовательности и в количествах, поставленных по различным маркам в различные ПП.

В нормальных ситуациях сходимость имитационного моделирования достаточно хорошая. В случае довольно сложной, реальной системы (6 ПП, из них 2 разветвления, 4 продукта - из них один только в два ПП) требуются 16-20 итераций для получения реализуемого плана.

Реализуемый план - эта совокупность временных диаграмм цикла событий в пункте снабжения, в пунктах потребления и в точках разветвления. Кроме того, конечно, можно выдать временные диаграммы запасов во всех резервуарах системы и необходимые итоговые данные.

Если имитационное моделирование после заданного количества шагов не сходится, можно перейти к моделированию следующего исходного плана поставок. В положительном случае также целесообразно моделировать и другие сохраненные исходные планы и проанализировать полученные решения. Для этой цели также имеются программы, которые рассчитывают такие показатели, как количество границ фаз (суммарные потери), наполненность резервуаров в конце цикла (причем "трудно-доступные" резервуары принимаются во внимание с большим весом), расхождение от количественного плана поставок, суммарное время цикла и т.д.

Имея реализуемый план поставок, целесообразно стремиться к его улучшению, локальной оптимизации. Целью может быть максимальное приближение к оригинальному количественному плану поставок, максимальное наполнение выбранных, "труднодоступных" резервуаров, уменьшение времени цикла и т.д. Эта задача решается методом линейного программирования, причем модель линейного программирования составляется автоматически по результатам имитационного моделирования.

4.3. Улучшение реализуемого плана поставок методом линейного программирования

Модель линейного программирования для улучшения реализуемого плана поставок состоит из трех основных частей:

- система уравнений материальных балансов по ПП и продуктам (по резервуарам, считая, что в каждом ПП под каждой маркой имеется единственный условный резервуар);
- система ограничений по интервалам времени и резервуарам, гарантирующих недопустимость опорожнения или переполнения резервуаров;
- система уравнений, характеризующих топологическую определенность системы.

При описании модели будем ссылаться на задачу поставки единственного продукта (п. 2.), потому что первые две части данной модели во многом сходны.

Необходимо уже в начале отметить, что в случае многопродуктовой модели искомые временные интервалы – в отличие от модели поставки газойла – являются интервалами между двумя событиями, а не временем одной операции (режима отбора продукта). Это приводит к увеличению количества временных интервалов, одна операция может длиться через несколько временных интервалов. В то же время конец некото-

рой операции и переход к следующей всегда является событием (манипуляция с заслонками), следовательно, началом нового временного интервала. Можно записать систему линейных уравнений для всех ПП и марок (для всех "резервуаров") аналогично системе уравнений (I):

$$\begin{aligned} v_{11}^1 \cdot x_1 + v_{12}^1 \cdot x_2 + \dots + v_{1s}^1 \cdot x_s &= b_1^1 \\ v_{11}^2 \cdot x_1 + v_{12}^2 \cdot x_2 + \dots + v_{1s}^2 \cdot x_s &= b_1^2 \\ v_{11}^j \cdot x_1 + v_{12}^j \cdot x_2 + \dots + v_{1s}^j \cdot x_s &= b_1^j \\ \vdots & \\ v_{n1}^m \cdot x_1 + v_{n2}^m \cdot x_2 + \dots + v_{ns}^m \cdot x_s &= b_n^m \end{aligned} \quad (6)$$

или в матричной форме

$$\underline{V} \cdot \underline{X} = \underline{B} \quad (7)$$

Отметим, что в данном случае матрица V – в отличие от матрицы v^* – берется только один раз, так как последовательность операций задана имитационным моделированием.

Если целью оптимизации будет приближение действительно поставленных количеств к первоначальному количественному плану, то записанные уравнения должны переписываться в неравенства, где одним из граничных значений будет действительно поставленное количество, другим – плановое количество. Всего получим $2 \cdot n \cdot m$ неравенств.

Вторая часть модели, система ограничений на запасы по резервуарам, полностью аналогична системе неравенств (2). Проблема заключается только в том, что количество таких ограничений велико: $2 \cdot n \cdot m \cdot s$, то есть в случае 5 ПП, 3 продуктов и 40 временных интервалов уже 1200 строк, что очень затрудняет решение задачи. Поэтому необходимо радикально уменьшить количество этих ограничений, что возмож-

но без ухудшения приемлимости полученного решения.

Перед составлением этой части модели анализируем ситуацию. Разумеется, что на переполнение следует проверить резервуары только после тех временных интервалов, в которых отбирался продукт в данный резервуар. Что касается опорожнения, здесь следует иметь в виду два момента. В связи с газойлом, уже упомянулось, что перед началом цикла многопродуктовой поставки создается такая ситуация, чтобы газойл нигде не кончался к концу цикла (цикл начинают с удовлетворением спроса по газойлу), следовательно, нет необходимости проверить опорожнение резервуаров под газойлом. Что касается остальных продуктов можно рассчитать момент ожидаемого опорожнения этих резервуаров и достаточно записать ограничения по их опорожнению только начиная с временного интервала ожидаемого момента опорожнения. В практике запишем эти неравенства начиная на два интервала раньше, так как в процессе решения модели временные интервалы могут сместиться (может увеличиваться время предыдущих временных интервалов). Еще один момент: после поставки планового количества (после последнего "обращения" к этому резервуару) нет необходимости продолжать проверку на опорожнение, так как плановое количество (и действительно поставленное количество, определенное имитацией, если оно меньше планового) обеспечивают достаточно пополнение запаса в резервуаре.

В результате такого несложного анализа можно значительно сократить количество ограничений на запасы в резервуаре. Если в одной операции можно отбирать продукт в не более два ПП, то максимальное количество ограничений на переполнение будет равняться $2 \cdot s$, то есть в случае приведенного примера не более 80. Количество ограничений на опорожнение зависит от конкретной ситуации, но лежит в этом же диапазоне. Размерность этой части модели будет вполне приемлимой.

Что касается третьей части модели, ее будем объяснять с помощью приведенного примера, возвращаясь к событиям имитационного моделирования. Суммарное время до наступления события № 3 определяется (при заданных интенсивностях отбора) объемом участка трубопровода А, то есть можно записать:

$$v_{12}^1 \cdot x_2 + (v_{13}^1 + v_{23}^1) \cdot x_3 = A \quad (8)$$

Время наступления события № 4 определяется объемом соединительного участка В и частично временем предыдущей операции:

$$v_{23}^1 \cdot x_3 + v_{24}^1 \cdot x_4 = B \quad (9)$$

Аналогично, временный интервал между событиями № 5 и 6 определяется объемом участка С:

$$v_{35}^1 \cdot x_5 = C \quad (10)$$

Именно это уравнения записываются в третью часть модели. Отметим, что временный интервал между событиями № 6 и 7 "свободен" от топологических ограничений: в системе имеется один продукт (бензин) и его нагнетание в ПП 1 и 2 не предопределяется топологией. Что касается временного интервала между событиями № 7 и 8, он снова определен объемом участка А и т.д.

Третья часть модели включает в себе именно уравнения типа (8), (9), (10).

На рис. 2. изображена модель для оптимизации приведенного примера.

Целевая функция записывалась следующим образом:

$$F = \sum_{i=1}^n \sum_{j=1}^m k_i^j \cdot q_{is}^j \rightarrow \max \quad (11)$$

или

$$F = \sum_{i=1}^n \sum_{j=1}^m k_i^j (b_i^j - b_{ia}^j) \rightarrow \min \quad (12)$$

где $k_i^j = 0$ если $b_{ia}^j \geq b_i^j$

Что касается третьей части модели, ее будем объяснять с помощью приведенного примера, возвращаясь к событиям имитационного моделирования. Суммарное время до наступления события № 3 определяется (при заданных интенсивностях отбора) объемом участка трубопровода А, то есть можно записать:

$$v_{12}^1 \cdot x_2 + (v_{13}^1 + v_{23}^1) \cdot x_3 = A \quad (8)$$

Время наступления события № 4 определяется объемом соединительного участка В и частично временем предыдущей операции:

$$v_{23}^1 \cdot x_3 + v_{24}^1 \cdot x_4 = B \quad (9)$$

Аналогично, временный интервал между событиями №№ 5 и 6 определяется объемом участка С:

$$v_{35}^1 \cdot x_5 = C \quad (10)$$

Именно это уравнения записываются в третью часть модели. Отметим, что временный интервал между событиями №№ 6 и 7 "свободен" от топологических ограничений: в системе имеется один продукт (бензин) и его нагнетание в ПП I и 2 не предопределяется топологией. Что касается временного интервала между событиями №№ 7 и 8, он снова определен объемом участка А и т.д.

Третья часть модели включает в себе именно уравнения типа (8), (9), (10).

На рис. 2. изображена модель для оптимизации приведенного примера.

Целевая функция записывалась следующим образом:

$$F = \sum_{i=1}^n \sum_{j=1}^m k_i^j \cdot q_{is}^j \rightarrow \max \quad (II)$$

или

$$F = \sum_{i=1}^n \sum_{j=1}^m k_i^j (b_i^j - b_{ia}^j) \rightarrow \min \quad (I2)$$

где $k_i^j = 0$ если $b_{ia}^j \geq b_i^j$

Вектор X		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
Уравнения материальных балансов	P_1^1	0	400	300	0	0	0	0	0	$= b_1^1$
	P_1^2	0	0	0	300	0	300	0	0	$= b_1^2$
	P_2^1	0	0	400	400	0	0	0	0	$= b_2^1$
	P_2^2	0	0	0	0	350	400	350	0	$= b_2^2$
	P_3^1	0	0	0	0	350	0	0	0	$= b_3^1$
	P_3^2	0	0	0	0	0	0	350	500	$= b_3^2$
Ограничения по запасам	q_{10}^1	$(400-z_1^1)$	0	0	0	0	0	0	0	$\leq M_1^1$
	q_{10}^1	$(400-z_1^1)$	$(300-z_1^1)$	0	0	0	0	0	0	$\leq M_1^1$
	q_{20}^1	0	$(400-z_2^1)$	0	0	0	0	0	0	$\leq M_2^1$
	q_{10}^2	0	0	$(300-z_1^2)$	0	0	0	0	0	$\leq M_1^2$
	q_{20}^1	0	$(400-z_2^1)$	$(400-z_2^1)$	0	0	0	0	0	$\leq M_2^1$

	q_{20}^2	0	0	0	$(350-z_2^2)$	$(400-z_2^2)$	$(350-z_2^2)$	0	0	$\leq M_2^2$

	q_{10}^2	$-z_1^2$	0	0	0	0	0	0	0	≥ 0
	q_{10}^2	$-z_1^2$	$-z_1^2$	0	0	0	0	0	0	≥ 0
	q_{30}^2	$-z_3^2$	$-z_3^2$	$-z_3^2$	$-z_3^2$	0	0	0	0	≥ 0
	q_{30}^2	$-z_3^2$	$-z_3^2$	$-z_3^2$	$-z_3^2$	$-z_3^2$	0	0	0	≥ 0
Уравнения топологии		0	400	$(300+400)$	0	0	0	0	0	$= A$
		0	0	400	400	0	0	0	0	$= B$
		0	0	0	0	350	0	0	0	$= C$
		0	0	0	0	0	0	350	0	$= B$
		0	0	0	0	0	0	350	500	$= C$
		1	1	1	1	1	1	1	1	$\leq T$

Примечания: 1. Допустимые операции по табл. 1.

2. P_1^j - № резервуара

Рис. 2. Модель для оптимизации плана многопродуктовой поставки

В случае целевых функций (II-I2) значения k_i^j могут использоваться в качестве "веса" доступности или важности резервуара.

Следует отметить, что модель линейного программирования имеет достаточно мало степеней свободы. Как показывали, большинство временных интервалов строго определено топологией системы, их можно только переместить по оси времени. В случае нашего примера единственным "свободным" интервалом является интервал нагнетания бензина в ПП1 и ПП2 (интервал между событиями № 6 и 7), что можно использовать для увеличения (или уменьшения) поставок в ПП1, ПП2 и ПП3. Увеличивать количество бензина, поставленного в ПП3, вообще нельзя: при данной последовательности операций оно определено отношением объема участков В и С и отношением интенсивностей отбора в ПП 2 и 3.

На основе изложенного можно сделать вывод, что улучшение плана поставок линейным программированием имеет смысл только тогда, когда пробки достаточно большие по сравнению с объемом продуктопровода (продолжительно имеют место такие ситуации, когда в продуктопроводе в целом или в закрытой автономной его части между пунктом снабжения и пунктами потребления этого участка имеется только одна фаза). Длинные разветвления также ухудшают положение ввиду того, что их опорожнение и повторное наполнение увеличивает количество предопределенных временных интервалов.

Отметим, что задача линейного программирования всегда имеет решение благодаря тому, что начальная ситуация - реализуемый план поставок - сама является решением.

Как упоминалось, спрос по газойлу удовлетворяется перед и после цикла многопродуктовой поставки. Можно объединить модель поставки газойла с моделью многопродуктовой постав-

ки и получить модель, описывающий полный цикл

перерыв – поставка газойла – многопродуктовая

поставка – поставка газойла – перерыв,

но в практике более целесообразно решать проблему поставки газойла сепарабельно.

5. Оценка предложенного решения задачи многопродуктовой поставки

Задача многопродуктовой поставки – как показывали – в общем виде исключительно сложная задача, составление адекватной модели уже связано с большими затруднениями и она практически нерешима для систем реальной сложности.

В то же время, вероятно нет смысла записать такую задачу в общем виде. На основе изложенного ясно, что в реальных **системах** и ситуациях мало степеней свободы. Последовательность пробок достаточно строго предопределена моментами ожидаемого опорожнения конкретных резервуаров и топологией системы. Как види, на основе заданной последовательности пробок можно составлять набор исходных планов поставок, но они с точки зрения целевой функции обычно мало отличаются друг от друга. Принципиальную проблему представляла бы только та редкая ситуация, когда проверенные исходные планы поставок нереализуемы и существует непроверенный, реализуемый план. В такой ситуации диспетчер задает другую последовательность пробок, другие плановые количества и этот процесс в реальных условиях обязательно сходится.

Ситуация совершенно аналогичная в случае модели линейного программирования, когда выяснилось, что степень свободы оп-ределения длины пробок может быть очень ограниченной и зависит от конкретной системы и ситуации. Могут быть системы и ситуации, когда вообще нет смысла локальной оптимизации полученного реализуемого плана.

Что касается значения принятых упрощений, адекватность предложенного решения с реальной системой, можно сказать, что предположение равномерного отбора в плановом периоде вполне допустимо ввиду того, что ПП являются достаточно большими буферами по сравнению с неравномерностью отгрузки (резервуары 2-20 тысяч м³, продукты отгружаются автотанкерами 10-20 м³). Скорость отбора продукта из продуктопровода, естественно, может быть меньше максимальной интенсивности отбора, определяемой сечением трубы. С одной стороны, предположение отбора максимальной интенсивности соответствует той цели, что максимально использовать мощность продуктопровода и снизить время цикла многопродуктовой поставки, в то же время могут быть ситуации, когда это может привести к преждевременному переполнению резервуара. В имитационное моделирование можно включить такую опцию, что в случае преждевременного переполнения снизить интенсивность отбора, но считать скорость отбора вообще переменной нельзя, потому что при этом практически нельзя задавать дискретные операции отбора.

Что касается ограничение на одновременный отбор (не более в двух-трех ПП), оно более-менее реальное. Можно допускать, конечно, одновременный отбор в четырех-пяти ПП с меньшей интенсивностью, но это не практикуется, потому что задача станет совершенно неконтролируемой для диспетчера. В случае нашей модели это приводит к значительному увеличению таблицы допустимых операций, увеличивается количество исходных планов поставок и незначительно усложняется имитационное моделирование. Необходимо отметить, что увеличение количества допустимых операций, например, разрешение одновременного отбора в трех ПП, увеличивает гибкость системы и может ускорить нахождение реализуемого плана.

6. Реализация

Пакет программ для решения задачи многопродуктовой поставка был реализован на ЭВМ IBM 303I на языке ПЛ-I. Для решения задачи линейного программирования применяли стандартный пакет фирмы IBM - ЕСЛ, что оказалось очень удобным и эффективным средством. Объем пакета программ (без ЕСЛ) - выше 4000 строк.

При решении реальных ситуаций (максимум: 6 ПП, 4 продукта) имитационная программа почти всегда нашла реализуемый план, для этого требовались 16-20 итераций. Что касается линейного программирования, там заполненность матрицы около 25 % (по статистике ЕСЛ). Эффект улучшения целевой функции обычно невелик: он составляло 5-13 %. Время решения задачи незначительная.

Работу пакета программы, варианты реализации заданного плана иллюстрируем на примере, близком к реальным задачам. Топология системы дана на рис. 3, допустимые операции - в табл. 2, исходные данные - постоянные и переменные - сведены в табл. 3. Результат имитационного моделирования - временные диаграммы 4 возможных реализаций плана - даются на рис. 4.

Литература

- [1] Инзелт П.: Задача снабжения региона нефтепродуктами при наличии продуктопровода для последовательной транспортировки нескольких нефтепродуктов. МТА SZTAKI Közlemények, 35., 7-26. Budapest, 1986.
- [2] Данциг, Дж.: Линейное программирование, его применения и обобщения. "Прогресс", М., 1966.
- [3] Yaohan Chu: Digital Simulation of Continous Systems. (Hung. Transl.), Műszaki Könyvkiadó, Budapest, 1977.
- [4] IBM Mathematical Programming System /370 (MPSX/370): Control Languages. Chapter 4: Extended Control Language Description. IBM, SH19-1094.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

n	- общее количество ПП (индекс i)
m	- общее количество продуктов (индекс j)
s	- общее количество временных интервалов (индекс k)
$v = \{v_{ik}^j\}$	- скорость отбора продукта в ПП (i -той ПП, j -тый продукт, k -той интервал времени), $\text{м}^3/\text{час}$
$\underline{Q} = \{q_{ik}^j\}$	- запас по ПП и маркам, м^3
q_{i0}	- запас по ПП и маркам в начальном моменте t_0 периода планирования T , м^3
$\underline{z} = \{z_i^j\}$	- потребление по ПП и маркам в период планирования T , $\text{м}^3/\text{час}$
$\underline{x} = \{x_k\}$	- интервал времени (между двумя операциями или событиями), час
$\underline{m} = \{m_i^j\}$	- объем резервуаров, м^3
$T = \{t_0, t_v\}$	- период планирования, час
$v = \{b_i^j\}$	- плановое транспортируемое количество в периоде T , м^3
$v^+ = \{b_{ia}^j\}$	- фактически транспортируемое количество по реализуемому плану поставок, определенному имитационным моделированием, м^3
w_{cs}	- максимальная пропускная способность продуктопровода, $\text{м}^3/\text{час}$
$\underline{w} = \{w_i\}$	- максимальная скорость отбора продукта по ПП, $\text{м}^3/\text{час}$

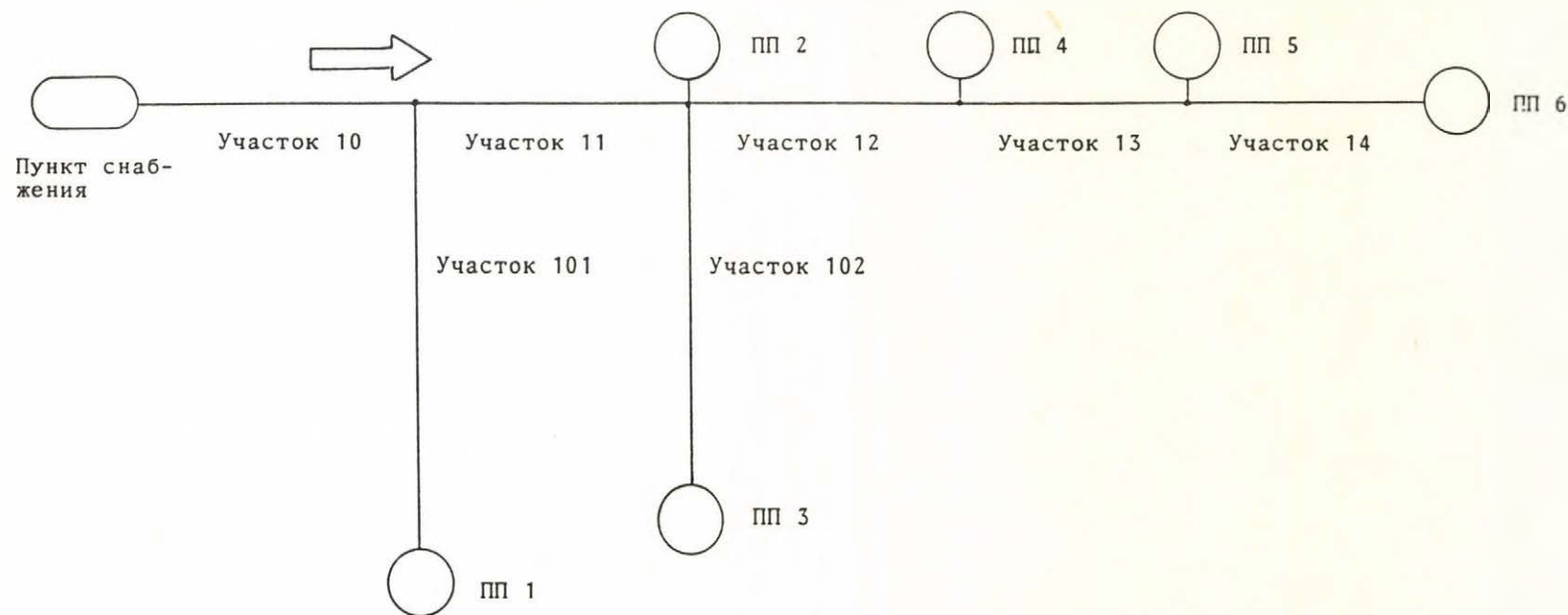


Рис. 3. Принципиальная схема образцового продуктопровода

Таблица 2.

Таблица допустимых операций, м³/час

№ операции	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
№ ПП																				
1	110	110	120	110	0	120	0	0	120	0	0	120	0	0	120	0	0	0	0	0
2	110	110	0	120	120	0	120	0	0	110	0	0	110	0	0	120	0	0	0	0
3	150	0	180	0	180	0	0	120	0	0	120	0	0	0	0	0	200	0	0	0
4	0	180	0	0	0	270	280	180	0	0	0	0	0	120	0	0	0	300	0	0
5	0	0	0	0	0	0	0	0	250	250	180	0	0	0	0	0	0	0	300	0
6	0	0	0	0	0	0	0	0	0	0	0	200	200	180	0	0	0	0	0	300

Таблица 3.

Исходные данные и результаты имитационного моделирования многопродуктовой поставки

№	№	№	№	№	№	Поставленные количества при различных вариантах плана поставки, м³				Запасы после окончания цикла при различных вариантах плана поставки, м³			
						1	2	3	4	1	2	3	4
1	1	1400	700	15	2500	2310	2372	2153	2310	965	987	757	965
	2	1000	500	6	1430	1263	726	1032	1002	946	395	696	684
2	1	1400	700	15	3200	2701	2289	2631	2731	1356	906	1234	1379
	2	1000	500	5	1650	1062	702	733	780	833	510	537	599
	3	1000	500	5	1500	517	591	834	400	339	399	638	219
3	1	10000	5000	20	2600	4431	4500	4431	4427	6702	6719	6634	6688
	2	7000	3500	6	1540	714	1436	1416	844	3398	4105	4080	3526
	3	1000	500	6	1400	937	887	864	912	623	556	528	594
4	1	100000	50000	40	3400	3580	3566	3589	3590	48131	48006	47996	48109
	2	20000	10000	6	1760	1654	1600	1600	1737	10839	10769	10764	10919
	3	10000	5000	5	1500	1500	1500	1500	1600	5821	5807	5803	5916
	4	30000	15000	0	0	0	0	0	0	15000	15000	15000	15000
5	1	11000	5500	39	4000	4190	4166	4190	4190	4367	4245	4235	4346
	2	8000	4000	7	1760	1618	1600	1600	1600	4666	4629	4623	4643
	3	8000	4000	6	1500	1552	1500	1500	1500	4736	4668	4664	4682
6	1	100000	50000	42	5000	5190	5077	5100	5100	49366	49238	49227	49344
	2	80000	40000	6	1650	1588	1500	1600	1860	40772	40669	40765	41041
	3	40000	20000	5	1300	1406	1300	1425	1370	20727	20608	20729	20687
	4	30000	15000	0	0	0	0	0	0	15000	15000	15000	15000
Время окончания цикла поставки, день/час										5/16.5	5/19.0	5/19.9	5/17.1

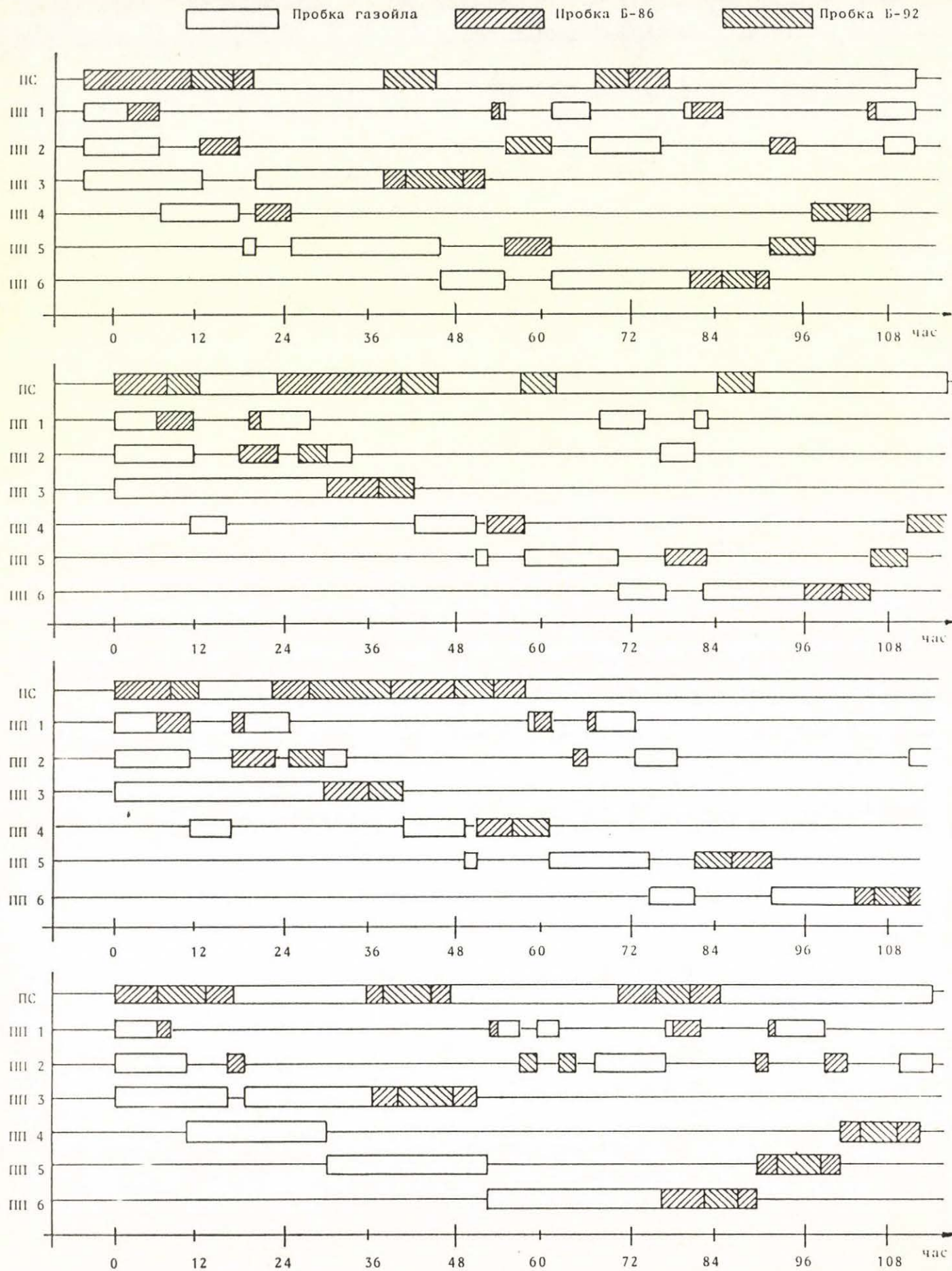


Рис. 4. Временные диаграммы 4-х реализуемых планов поставок, полученных имитационным моделированием

БАЗОВЫЕ ЭЛЕМЕНТЫ В АВТОМАТИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ АДМИНИСТРАТИВНО-УПРАВЛЕНЧЕСКИХ СИСТЕМ

Румяна К.Киркова, Юлиана П.Каравасилева
Институт математики с ВЦ Болгарской АН

Введение

Тенденции развития средств обработки данных и информационного обслуживания в настоящее время таковы, что информационно-вычислительные ресурсы все более доступны для конечных пользователей как в логическом плане, так и с точки зрения их физического распределения по структуре организации или учреждения.

Производительные и экономичные микропроцессоры создали возможность значительно приблизить устройства обработки данных к рабочим местам администраторов, финансовых работников, руководителей управления и т.п.

Поэтому современный этап автоматизации информационных процессов (ИП) определяется возможностями новых элементов информационной технологии, к которым следует отнести:

- распределенные базы данных;
- интеллектуальные терминалы;
- профессиональные микро ЭВМ;
- локальные вычислительные сети

и др.

На основе таких элементов строятся прикладные модули для автоматизации ИП административно-управленческих систем (АУС), специализированные по различным сферам учрежденческой деятельности.

Архитектурные принципы АУС

АУС характеризуются рядом специфических свойств:

- потребителями АУС являются административный и технический персонал, руководители проектов,

и подразделений, экономисты и финансовые работники, библиотекари и специалисты по информационному анализу, научные работники;

- управление и эксплуатация прикладных модулей АУС, как правило, осуществляются самими пользователями систем; терминалы, рабочие станции, устройства коммуникации и отображения информации устанавливаются непосредственно в рабочих местах и доступны для использования в любое время;

- обработка информации выполняется на терминалах, микро-, мини- или больших ЭВМ, объединенных общей локальной вычислительной сетью учреждения; кроме того имеется доступ к удаленным ЭВМ при помощи межсетевых переходов к глобальным вычислительным сетям;

- пользователи систем изолированы от сложной многоступенчатой структуры систем обработки и передачи информации; каждый прикладной модуль связан с пользователем верхним слоем функций и управляющих команд; эти функции и команды представляются в простой и понятной форме (например типа "меню" и т.п.);

- пользователям дается широкий спектр информационных и вычислительных услуг, отвечающий рабочим функциям многих категорий сотрудников, все услуги доступны и активны в каждый момент времени.

Наличие совокупности указанных свойств означает, что информационно-вычислительные ресурсы учреждения интегрированы в единое целое. Разделение услуг по подсистемам или видам обслуживания остается в планах развития систем. Конечные пользователи в своей практической работе рассматривают эти возможности как взаимодополняющие инструменты учрежденческой деятельности.

АУС интегрируют средства доступа пользователей как к внутренним информационно-вычислительным службам (как например электронная почта, ввод

и хранение информации в баз данных, электронный календарь, текстовые обработки, копирование и распределение документов, машинная графика, телеконференции, обработка и передача звуковой информации, средства для автоматизации программирования и проектирования, средства обеспечения принятия управленческих решений и т.д.) , так и к внешним информационным источникам(как например телексная или теле-тайпная связь, электронные газеты, услуги глобальных сетей ЭВМ и т.д.).

Естественно, что различные категории пользователей по-разному нуждаются в обращении к внутренним и внешним информационным службам.

Критерии проектирования ИП АУС

Беря за основу высказанные принципы можно сформулировать ряд критериев, которым должны удовлетворять ИП АУС:

- релевантность: получаемая информация должна быть релевантной запросам любого лица, принимающего решения; эта информация различается в зависимости от потребителей, которыми могут быть либо директор организации, либо техническое лицо. Существует ключевое различие между данными и информацией. Информация – это данные, релевантные потребителю;

- управление по направлениям: административному работнику не требуется большого количества детальной информации. Он должен хорошо быть осведомлен о критических факторах, влияющих на успех организации;

- точность: данные, на основе которых формируется информация, должны быть правильны, с тем что-бы решения принимались на основе точной информации, требуемой для рассматриваемой задачи. Точность также предполагает, что информация отражает текущее состояние, и следовательно, она не должна основываться на устаревших данных;

- современность: информация должна предоставляться именно тогда, когда она необходима потребителю;

- приспособляемость: система должна быть способна удовлетворять различные запросы пользователя. Эти запросы могут принимать форму контроля итоговых сумм или быть более детализированными. В системе должны быть также предусмотрены возможности перепроектирования в соответствии с изменяющимися потребностями пользователей.

Требования при проектировании ИП АУС

Надо отметить, что иногда чтобы удовлетворить вышеуказанные критерии, проектировщики информационных услуг АУС сталкиваются с следующими, в некотором смысле противоречивыми, требованиями:

- чем больше, тем лучше: управленческому персоналу часто предоставляется информация больше, чем ему требовалось. Данные следовало бы проверить на релевантность и концентрировать их в соответствии с действительными потребностями. Иногда например следовало бы давать контрольные суммы, а не детальные данные — такое требование называется "управление по сводным данным";

- управленческий персонал нуждается во всей той информации, которую он запрашивает: администраторы систем часто впадают в искушение запросить всю доступную информацию. Системный аналитик должен гарантировать, что потребности в информации будут удовлетворяться только после тщательного исследования процесса принятия решений. Информация не должна представляться только по той причине, что она доступна. Поэтому системному аналитику необходимо знать о действительных потребностях администратора;

- процесс принятия решений способствует развитию представляемой информации: администраторы могут фактически игнорировать получаемую информа-

цию, хотя она и является релевантной, полагаясь только на свою интуицию, опыт и **знании**. Они должны знать потенциальные возможности системы, и им должно быть гарантировано эффективное использование этих возможностей;

- повышение коммуникабельности ведет к росту эффективности работы: администраторы, работающие с АУС, могут иметь доступ к информации, относящейся к другим подразделениям или подсистемам организации. Это в конце концов может привести к нежелательным конфликтным ситуациям между подразделениями; подобные последствия должны быть предусмотрены до того, как будет разрешено свободное пользование информацией;

- администратор должен только уметь использовать систему: для того чтобы иметь возможность оценивать систему и осуществлять управление ею, администратору необходимо обладать некоторыми знаниями о механизме системы. В связи с этим необходимо развивать взаимодействие между пользователями, с одной стороны, и системным персоналом и ЭВМ с другой. Без этого невозможно установить дух взаимного доверия и кооперации. В наиболее передовых организациях инициативе пользователей придается первостепенное значение.

Базовые элементы в АУС

Концепция "базовые элементы в АУС" позволяет выстроить полную иерархическую цепочку элементов АУС

базовый элемент \Rightarrow прикладная подсистема
 \Rightarrow услуга прикладной подсистемы \Rightarrow прикладной процесс

Это объясняется тем, что АУС образуется совокупностью прикладных процессов, управляющих сетевых служб, протокольных служб различных уровней, прикладных интерфейсов пользователей, взаимодейст-

вующих в целях:

- решения комплекса прикладных задач пользователя в рамках АУС;
- интеграция распределенных физических элементов системы;
- улучшения условий труда (т.е. эргономика рабочих мест, повышения квалификации, интеллектуализация труда и т.д.).

В большинстве известных проектов АУС реализуются принципу взаимодополняемости персональных инструментов учрежденческой работы и прикладных подсистем АУС. Для них характерно использование вполне определенных управленческих технологий, баз данных, регламентов решения задач, исполнителей. Для персональных инструментов наиболее важны свойства привычности и полезности для конечных пользователей, произвольности решений по их использованию, субъективности в оценке их эффективности.

Прикладную систему образует набор решаемых задач и конструкционные блоки, которые используются для этих задач. Состав решаемых задач, даже для учреждения среднего масштаба, может превышать порядка 100, поэтому целесообразно их объединение в прикладных подсистем. Типичным составом таких подсистем для научно-производственной организации является:

- научно-тематическое планирование,
- управление проектами,
- текущее планирование,
- финансовое планирование и экономический анализ проектов,
- материально-техническое снабжение,
- подготовка, оформление и контроль выполнения договоров,
- оперативное управление и контроль заданий,
- информационно-библиотечное обслуживание,
- набор отчетов, докладов и публикаций,

- редакционная подготовка и внутренний документооборот,
- кадровая обеспеченность во всех направлениях,
- научные связи с другими организациями (научно-тематические конференции, семинары, совещания и т.п.),
- подготовка корреспонденции,
- электронная почта.

Отметим, что подсистема "электронная почта", как и ряд других подсистем, имеет двойственное назначение - как самостоятельная коммуникационная подсистема и как прикладная обеспечивающая подсистема.

Для решения вышеупомянутых задач используются также общесистемные средства:

- технические средства передачи и обработки данных,
- рабочие станции,
- ЭВМ разного типа,
- программные средства управления обменом информацией,
- операционные системы ЭВМ.

Таким образом АУС можно рассматривать в двух аспектах: прикладном (учрежденческом) и сетевом. Подход к анализу и синтезу АУС как к сетевой распределенной системе, ориентированной на решение базовых учрежденческих задач, имеет важное практическое значение, поскольку обосновывает стратегию создания и развития АУС:

- первоочередное решение коммуникационных задач в качестве основы подсистем "электронная почта", а также задач других базовых элементов;
- автоматизация решения задач базовых элементов "обработка информации", "принятие решений", "управленческие процедуры" с параллельной интегра-

цией соответствующих прикладных подсистем на базе локальной вычислительной сети;

— разработка компоненты базового элемента "персональные инструменты" на основе языков высокого уровня и других средств персональных ЭВМ.

Персональные ЭВМ в ИП АУС

Техническое развитие персональных ЭВМ исключительно динамично. Только в течении последних 10 лет в мире разработаны уже три поколения:

- 8-битовые микропроцессоры с объемом оперативной памяти до 64 Кбайтов;
- 16-битовые микропроцессоры с объемом оперативной памяти выше 1 Мбайта что значительно расширяет сферу их применения;
- 32-битовые микропроцессоры с очень развитыми вычислительными мощностями.

В следующих годах ожидается, что их технические возможности будут развиваться таким же темпом.

Как было упомянуто, персональные ЭВМ могут служить основной компонентой при автоматизации ИП в АУС. Уже существует большое количество разных прикладных программ и систем:

— для создания и управление логически несвязанных отдельных файлов: здесь связь с пользователем осуществляется в простой и понятной форме (типа "меню", но эти системы не очень гибкими по отношению требований пользователя. Типичным примером является система PFS ;

— для одновременного поиска по нескольким показателям в нескольких файлах: эти системы более гибкими, так как они предоставляют возможности наращивания путем применения специализированным программных языков. Типичным примером — система DBASE-III ;

— для создания и управления базами данных: эти системы предоставляют очень большие возможно-

сти для информационного обеспечения, но за то требуют более значительные вычислительные ресурсы. Как представителем отметим система DBASE-III для создания реляционных баз данных.

В последнее время большой популярностью пользуются так называемые интегрированные программные системы, состоящих из нескольких модулей использующих общую базу команд. Таким способом они реализуют по нескольким базовым функциям, как например текстообработка, машинная графика, управление файлами и базами данных. Этим объясняется их широкое развитие и применение в настоящее время. Но нельзя забывать, что наращивание возможностей данной системы всегда требует более развитую техническую базу.

Заключение

Подводя итоги рассмотрения принципов построения АУС следует обратить внимание на следующие вопросы:

- 1) Можно ли понимать создание АУС лишь как постепенное насыщение вычислительной техникой и оргтехникой всех участков учреждения?
- 2) Завершается ли создание АУС автоматизацией всех действующих и в большой мере традиционных учрежденческих технологий?
- 3) В какой степени коренные эффекты АУС определяются внедрением управленческих процедур нового типа, обладающих повышенной информационной сложностью?
- 4) К каким эффектам АУС следует стремиться в первую очередь: к снижению трудозатрат персонала на обработку информации или к повышению обоснованности принимаемых управленческих решений?

Ответы на эти вопросы непосредственно определяют стратегические принципы построения АУС. В

настоящий момент развития можно говорить о важности и перспективности таких стратегических направлений создания АУС, которые обеспечивают:

- интеграцию различных видов коммуникаций, средств оргтехники, обрабатывающих систем с целью создания удобных, гибких и функционально расширяемых прикладных систем;

- взаимопроникновение в управленческие технологии тщательно разработанных, формальных процедур и других существующих неформальных систем межперсональной коммуникации и личной работы;

- аккумуляцию опыта применения новых технологий АУС и активное воздействие этого опыта на развитие эргономики рабочих станций, интерфейсов пользователей с системами, общих организационных условий внедрения подсистем АУС;

- постоянное наращивание логической сложности методов и процедур АУС, опережающих развитие подсистем подготовки и принятия управленческих решений.

Таким образом, на пути к будущим "электронным учреждениям" предстоит решить ряд серьезных научных и практических задач, отвечающих высоким требованиям современного этапа научно-технического прогресса.

ЛИТЕРАТУРА

1. Глушков Б.М. - Основы безбумажной информатики, Москва, Наука, 1982.
2. Кроув Т., Эйвисон Д. - Базы данных в административных информационных системах, Москва, Финансы и статистика, 1983.
3. Тенев И. и др. - Автоматизирани учрежденски системи, София, Техника, 1983.
4. Програмни продукти за автоматизация на учрежденийката дейност, София, БСНИИ "Интерпрограма", 1985.

НЕКОТОРЫЕ ВОПРОСЫ О СОЧЕТАНИИ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЛОКАЛЬНОЙ СЕТИ НА БАЗЕ МИКРО-ЭВМ

Ле Мань

Институт Вычислительных наук и Кибернетики

На развитие современной мировой техники большое внимание оказывают "Революционные" микрокомпьютеры и их математического обеспечения /МО/ в вычислительных науках. Подход соединения микро-ЭВМ между собой для создания локальной сети /ЛС/ является актуальной проблемой. Структура этих ЛС базируется на модели связи межпроцессных взаимодействий. В настоящее время имеются некоторые типы ЭВМ: Универсальные, мини и микро. Каждый из этих типов имеет свое МО. Следует отметить, что при создании ЛС из различных ЭВМ, важнейшей проблемой является сочетание их МО. Чтобы выполнить эту задачу, предлагаем использовать кросс-систему, которая переводит команды и приказы МО одной машины в команды МО другой машины. Эта кросс-система является частью протокола прикладных уровней в семи взаимонезависимых уровнях взаимодействующих открытых систем /ВОС/. Данная работа посвящена этим подходам.

1. Общие принципы кросс-системы

Многие фирмы уделяют вопросам разработки кросс-средств большое внимание. Эти средства разрабатываются, как правило, на универсальных или на мини ЭВМ с развитой МО. Микро-ЭВМ, если они используются для создания кросс-средств, должны быть совместимы с теми микро-ЭВМ, для которых эти системы разрабатываются. В 1976 году на Евромикросимпозиуме в Вене было доложено о программе CASS 8, которая является кросс-системой, написанной на языке высокого уровня /Фортран/ для обучения студентов. Эта система реализована на большой ЭВМ CDC-6600, предназначена с различными микро-ЭВМ на ассемблер-

ном уровне и уровне машинных кодов этих машин. Программа CASS 8 использует свойства инструментальной машины:

- 1/ Коды операций 8-разрядных микро-ЭВМ содержат сведения не только о характере операции, но и о том, к какому регистру она относится, каким будет способ адресации операнда и константы.
- 2/ Для операнда возможны четыре формата:
 - шестнадцатиричный код,
 - десятичный код,
 - код ASCII,
 - код символы.
- 3/ Если операнд, написан в виде символьного кода, то после него, может быть указан признак старшего байта /H/ или младшего /L/ для 16-битного индекс-регистра.
- 4/ Программа CASS 8 допускает девять псевдоинструкций отражающих специфику машины-хозяина, в их числе оператор начала и конца программы, оператор присваивания, оператор установки счетчика команд, оператор резервирования ячеек памяти, оператор данных, два оператора, позволяющих иметь сведения о машинном времени, затраченном на выполнение программы.
- 5/ Сведения о машине, для которой пишется программа, заводятся в виде файла инструкций.

Результатом работы программы CASS 8 является программа в кодах соответствующей МИКРО-ЭВМ. Эта полученная программа затем вводится в микро-ЭВМ.

Быстрое развитие микро-ЭВМ и микропроцессорной техники вызвало необходимость автоматизировать процесс отладки и регулирования новых машин. На базе микро-ЭВМ с развитыми системами МО, создаются генераторы ассемблеров и симуляторов для микропроцессоров.

В лаборатории технической кибернетики Вычислительного цент-

ра АН СССР имеется универсальная кросс-система, реализованная на микро-ЭВМ "Электроника-60" /Э-60/, предназначенная для программирования различных микро-ЭВМ с байтной организацией памяти и данной командного слова не более 16 битов.

В системе в качестве входного языка принят язык ассемблер Э-60, допускаются макрокоманды, определенные предварительно. Основные проблемы, возникающие при создании системы, использующейся в качестве инструментальной машины Э-60:

- 1/ Э-60 имеет фиксированное командное слово, все адреса команд четные, несовместимыми могут оказаться также и системы представления чисел.
- 2/ Со стороны целевых машин основной проблемой является ввод новых способов адресации и кодов операций, отсутствующих в ассемблере инструментальной машины. Исходная программа, написанная для конкретной микро-ЭВМ /здесь система реализована для некоторых микро-ЭВМ: "Электроника С5", микро-ЭВМ с центрального процессора MOTOROLA-6800 и INTEL 8080/ в ассемблере Э-60 с допустимыми макрокомандами предьявляется программе Предассемблер, в задачу которой входит перетрансляция макрокоманд в ассемблер Э-60. Для ввода новых кодов операций и способов адресации принята система макровывзовов подпрограмм, работающих на уровне имитатора. В программе пользователя такой вызов замещается командами программ много прерывания /EMT /n/ и TRAP /n/ /. Байтный номер этих команд является либо вводимым кодом операции, либо может быть поставлен с ним в таблицах соответствия на уровне кросс-системы. Основным ядром системы является компилятор "Коды-Коды" /ККК/. Он делает ссылки на таблицы соответствия команд, которые вводятся для каждой конкретной микро-ЭВМ при генерации системы. Данные о конкретной машине вводятся с помощью программы-диалога /см. на рис. 1./.

В настоящее время в Вычислительном центре АН СССР эта программа была введена в эксплуатацию.

"Инструментальная" машина в кросс-системе выполняет функцию преобразования прикладных программ по некоторому принципу, позволяющему переводить их на соответствующие программы на других языках.

Метод создания универсальной кросс-системы позволяет расширить вычислительные способности других машин с помощью инструментальной машины.

2. Алгоритмы и реализация двухсторонней кросс-системы для ЛС на базе разнородных микро-ЭВМ

Под односторонней кросс-системой подразумевается система, реализованная на инструментальной машине с выходными программами на языках подчиненных машин. Эта система лишь расширяет вычислительную способность подчиненных машин. Однако, если создается двухсторонняя кросс-система, то вычислительная способность каждой машины расширяется благодаря сочетанию МО двух машин.

Двухсторонняя кросс-система - это система содержащая в себе две подсистемы - "прямую" и "обратную" кросс-системы. "Прямая" кросс-система реализуется на центральной /инструментальной/ машине, а "обратные" кросс-системы создаются на подчиненных машинах.

Нами была реализована в настоящее время двухсторонняя кросс-система для ЛС, созданная на основе двух типов микро-ЭВМ LSI-11 и APPLE-2 /прямая кросс-система реализована на LSI, а обратная - на машине APPLE-2/. Микро-ЭВМ играет роль центральной машины и LSI, а машина APPLE-2 - подчиненных машин. Прямая кросс-система переводит ассемблеры LSI с ограничением в ассемблер APPLE-2. Она позволяет транслировать некоторые программы нужные для обработки графики высокой плотности в программы APPLE-2 /из-за машины LSI еще не имеет устройства обработки графики/. Обратная кросс-система позволяет перевести всю систему команды APPLE-2 на соответ-

ствующие команды LSI. Выходные программы поступают в каждую машину через каналы связи. Так как каждая машина работает под различной операционной системой, /для LSI на ОС RT-II, а на APPLE-2 DOS Appsoft/ поэтому в кросс-системе должен иметься эмулятор операционных систем, этот эмулятор позволяет перевести команды управления одной машины в другую и обратно.

Например: на APPLE имеет команду "CATALOG", то через эмулятор ОС будет перевести на выходной программы "DIR".

Рассматриваются пути преодоления трудностей, возникающих из-за программной несовместимости между двумя машинами в ЛС, вследствие различной организации регистров в центральных процессорах различного представления информации /чисел и команд/, наличия специфических способов адресации и т.п. Исследуются принципы построения предассемблера, дающего возможность вводить новые выражения входного языка. Рассматривается способ генерации формализованных описаний различных микро-ЭВМ в кросс-системе в виде специальных признаков и таблиц соответствия кодов операций и способов адресации. Генерация производится в форме программы /см. на рис. 2./.

Приведем краткую характеристику ЛС на двух типах микро-ЭВМ:

- ЛС с модулем центрального управления "Хозяйка-Подчиненны".
Операционная система управления сети реализована на машине LSI.
- Структура ЛС - ВОС.
- Низкие протоколы используются V24 /для физического уровня/ HDLC /для канального уровня/ и X25 /для сетевого уровня/.

Технические характеристики для каждой машины:

Для машины LSI-II:

- ее операционная система в реальном масштабе времени /RT-II/, возможно управлять ОЗУ в режиме виртуальной,

- объем ОЗУ 64 Кслов,
- стандартная клавиатура и дисплей,
- матричное печатающее устройство со 132 символами в строке,
- два накопителя на гибких магнитных дисках /НГМД/. Общая емкость одного диска с размером 8 дюйм - 0,6 Мбайтов,
- программное обеспечение имеет компиляторы для языков Фортран, PASCAL, интерпретер для Бейсика, макроассемблер и ассемблер.

В комплексе ПО тоже имеются пакеты прикладных программ, которые ориентированы на различные области применения.

Для машины APPLE-2:

- операционная система - DOS applesoft,
- объем ОЗУ 64 Кбайта,
- имеет плату Z-80, разрешит машине работу ОС - CP/M 80,
- стандартная клавиатура и дисплей с возможностью графики высокой плотности,
- матричное печатающее устройство с 80 символами в строке,
- две НГМД с емкостью одного диска размером 5 1/4 дюйма 92 Кбайта,
- программное обеспечение: компиляторы Фортран microsoft, Pascal UCSD, интерпретатор Бейсик, Бейсик applesoft, мини ассемблер и ассемблер. Пакеты специальных программ VISICAL, MULTIPLANC WORD PROCESSOR ...

3. Некоторые открытые проблемы:

Хотя проблеме трансляции языков с одной машины на другую уже давно было уделено внимание, но она еще требует усовершенствования.

Эксплуатация созданных кросс-систем показывает следующие проблемы:

1. Кросс-система будет эффективной, если заранее знают структуру транслируемых программ, адреса памяти хранения управляющих параметров и данных или по крайней ме-

ре предсказывают указанные элементы.

2. Кросс-система хорошо работает с прикладными программами не очень сложной структуры.
3. Использование кросс-системы для трансляции системой программы /пакетные или библиотечные программы/ имеют следующие трудности:
 - а/ Адреса буферных ячеек и адреса ОЗУ, хранящие управляющие параметры, могут быть транслируемы, хотя эти ячейки не нужно транслировать. Это будет нарушать структуру программы.
 - б/ Кросс-система всегда считает управляющие и данные параметры программы командами.
 - в/ Объем памяти транслированной программы не соответствует объему первоначальной программы.

Надеемся, что указанные трудности при трансляции системой программы частично будут решены нами в ближайшее время.

Созданные нами кросс-системы были рассмотрены и опробованы только на низком уровне языков ассемблера. Но по-нашему замечанию, лучше было бы создать кросс-систему позволяющую перевести промежуточные коды машин друг в друга. Это позволяло бы перенести с одной машины в другую программы, написанные на языках высокого уровня. К сожалению, до сих пор у нас не имелось описание промежуточных кодов машин, поэтому такая кросс-система еще не создается в нашей стране.

4. Заключение

Сочетание в ЛС различных микро-ЭВМ является необходимой работой в нашей стране. Выше была изложена попытка связывания в систему машины в ЛС путем использования кросс системы.

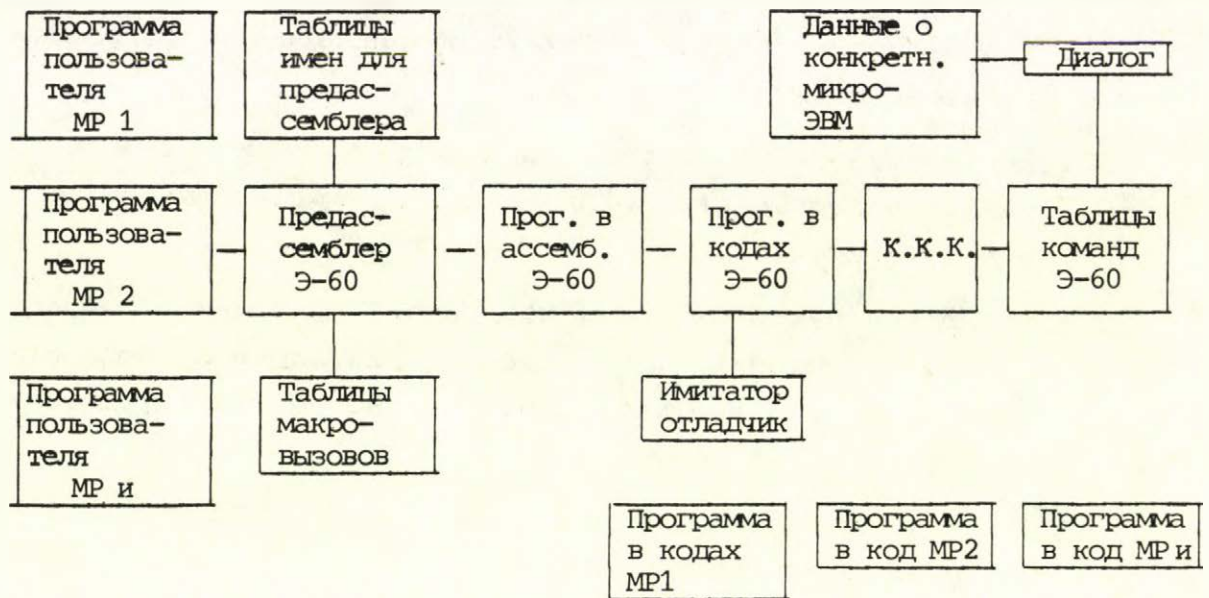


Рис. 1. Блок-схема универсальной односторонней кросс-системы.

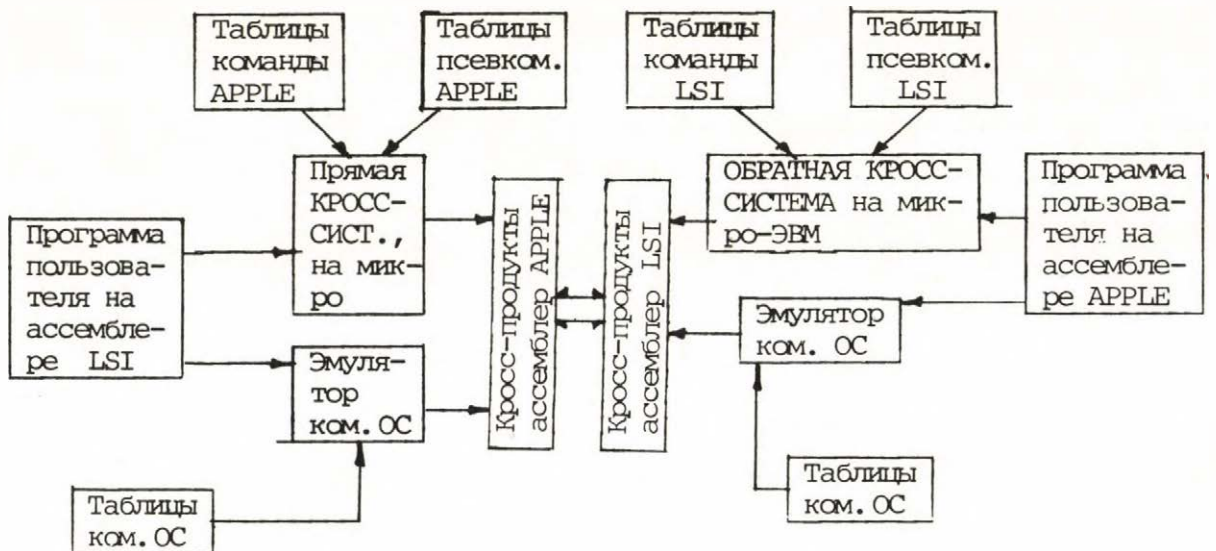


Рис. 2. Блок-схема двухсторонней кросс-системы.

Л и т е р а т у р а

1. Ю.И. Торгов и Ле Мань: Универсальный кросс-ассемблер для микропроцессорной системы. В трудах Всесоюзной конференции "Диалог-82" Индивидуальные системы на базе ЭВМ, гор. Пушкино, 1982 г.
2. Ле Мань: Комплекс средств подготовки системного и прикладного программного обеспечения для мультимикромашинных систем на базе разнородных микро-ЭВМ. Диссертация на соиск. ученой степени кандидата технических наук. г. Москва, 1982 г.
3. Ле Мань и Фам Нгос Хной: Некоторые вопросы об операционной системе для ЛС на базе микро-ЭВМ. Журнал "Наука и Техника для автоматического управления" стр. 18-35, №2-3, 1984 г. /Вьетнамский язык/.
4. Ле Мань: Методы повышения разработки информации для АСУ через создание Локальной сети на базе микро-ЭВМ. В трудах конференции "Применение математики, кибернетики и информатики для экономического управления" стр. 50-52, т. 2, Ханой 4/1984.
5. Ле Мань: О сочетании математического обеспечения ЭВМ через кросс-систему. В трудах 3-ей конференции Математики Вьетнама, Ханой 7/85.
6. Журнал ТИИЭР, том 71, № 12, Декабрь 1983.

СИСТЕМА АВТОМАТИЗИРОВАННОГО ОБСЛУЖИВАНИЯ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ

Марков Кр. К., Институт математики БАН

Большое значение информационного взаимодействия в настоящее время, его сложность и интенсивность даже в небольших социальных группах, требуют развития его автоматизированного обслуживания на базе комплексного подхода. Один такой возможный подход обсуждается в настоящей работе.

1. ВВЕДЕНИЕ

Рассматриваемый ниже комплексный подход к обслуживанию информационного взаимодействия /ИВ/ является решением следующих задач:

- определить формально основные понятия, связанные с ИВ;
- изучить свойства ИВ с математической точки зрения;
- определить какие стороны ИВ возможно обслуживать автоматизированно и какие нельзя обслуживать;
- описать основной базис обслуживающих операций, которые возможно и не принадлежат самому ИВ;
- разработать специализированный язык, с помощью которого обслуживать ИВ;
- построить систему автоматизированного обслужива-

живания ИВ /САОИВ/ и реализовать с ее помощью конкретные системы, которые на основании психологических наук индивидуализировать в соответствии с принятыми формами ИВ в конкретных типах социальных формаций.

Этот подход можно рассматривать как попытка "вынести за скобки" часто встречающиеся функции обслуживания ИВ, а в "скобках" оставить функции, касающиеся как процесса, ради которого осуществляется ИВ, так и специфических форм ИВ, присущих только некоторым формациям.

В этой работе рассмотрены некоторые результаты, связанные с этим подходом. Они являются следствием применения концепции информационного обслуживания, разработанной под руководством проф. П. Барнева в Институте математики Болгарской академии наук [1, 2].

2. ИНФОРМАЦИОННОЕ ВЗАИМОДЕЙСТВИЕ

В ходе своей деятельности субъект /человек, группа людей/ отражает в своем сознании окружающую реальность /материальные объекты/ с помощью некоторых рецепторов. Отражения в сознании субъекта назовем "идеальные объекты". При необходимости субъект материализует некоторые из находящихся в его сознании идеальных объектов в окружающую реальность /материальные объекты/ /рис.1./.

Очень часто, субъекту необходимо взаимодействовать с другими субъектами, с помощью которых

возможно достижение некоторых целей субъекта. Чтобы осуществить взаимодействие, субъект материализует отдельные идеальные объекты и таким образом делает их доступными для восприятия /рис.2./. Такие материализации, сделанные с целью взаимодействия назовем "информационные объекты" /ИО/. С информационными объектами можно проводить некоторые операции: перенесение в пространстве и времени, уничтожение и др. Материализация и отражение информационных объектов тоже будем понимать как операции.

Любую последовательность операций с информационными объектами назовем "информационный процесс".

Совокупность информационных процессов, проводимых с целью обеспечения взаимодействия субъектов назовем "информационное взаимодействие" /ИВ/.

Каждая подсовокупность ИВ назовем "неполное информационное взаимодействие" или короче - "информационный контакт" /ИК/.

Совокупность информационных объектов, используемые субъектами для ихнего ИВ назовем "информационная база" /ИБ/.

Иногда взаимодействующие субъекты не в состоянии сами достаточно качественно провести ИВ. Тогда целесообразно использовать постороннюю помощь. Совершение информационных процессов с целью облегчения реализации какого-нибудь ИВ будем называть "обслуживание" этого ИВ. "Автоматизированное обслуживание" ИВ совершается человеком с использованием ЭВМ, т.е. "обслуживающей системой человек-ЭВМ" /SIS/ /рис.3./.

3. СИСТЕМА АВТОМАТИЗИРОВАННОГО ОБСЛУЖИВАНИЯ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ /САОИВ/

Рассматриваемая здесь САОИВ основана на множестве формальных, программных, технических и методологических элементов. Она построена на базе мини-ЭВМ СМ-4 и операционной системы ДОС-РВ-Б. Программные части САОИВ написаны на макроассемблерном языке "МАКРО". Определяющими для САОИВ являются:

- многообластная информационная модель;
- многообластной метод доступа;
- система информационного контакта;
- методология работы с системой;
- операционная среда функционирования.

3.1. МНОГООБЛАСТНАЯ ИНФОРМАЦИОННАЯ МОДЕЛЬ /МИМ/

Суть МИМ состоит в том, что ИБ рассматривается как четырехмерное дискретное пространство /рис.4./. В каждой точке этого пространства может находиться или нет ИО. Где в этом пространстве поместится данный ИО зависит от субъектов ИВ. Если в данной точке нет ИО, то предполагается что в этой точке имеется стандартный "пустой" ИО.

Поскольку порождение ИО в данной точки, его существование и его устранения из этой точки протекают во времени, то одно из четырех измерений пространства ИБ является время.

Остальные три измерения вводятся согласно следующих простых соглашений:

- первое измерение используется для указывания отдельных объектов и называется "объекты":

- второе измерение используется для указания групп объектов и называется "области". В сущности, если представить себе "плоскость" из ИО, то все воображаемые прямые, параллельные "объектной" оси формируют отдельные области /рис.5./.

- третье измерение используется для указания групп областей и называется "архивы". По существу это соответствует обычному значению понятия архив. В модели архивы можно представить как параллельные плоскости /рис.6./.

На базе так введенной координатной системы возможно описывать связи между ИО в ИБ. Описания связей между ИО названы "индексы". Индексы можно поместить в ИБ в любую точку.

Иногда данный ИО из ИБ возможно "декомпозировать", т.е. рассматривать тоже как пространство из подобъектов с соответствующей четырехмерной координатной системой /рис.7./.

Естественно, допустимо и обратное рассмотрение - "композиция" множества ИО в одном ИО.

Для четырех основных структур ИММ /ИО, область, архив, ИБ/ допустимы следующие основные операции:

- создание;
- модификация;
- сравнение;
- декомпозиция;
- композиция;
- уничтожение.

Отметим, что операции "создание" и "уничтожение" являются особыми вариантами операции "модификация" /учитывая существования "пустых структур/.

Описание связей между структур, т.е. создание соответствующих индексов, можно рассматривать как особый вариант операции "композиция". Соответственно, оперирование с структурами следуя некоторого индекса можно рассматривать как особый вариант декомпозиции виртуального ИО, описываемого этим индексом.

3.2. МНОГООБЛАСТНОЙ МЕТОД ДОСТУПА /ММД/

Логическая организация ММД полностью соответствует МИМ. Ограничения сводятся единственно до размеров структур МИМ.

ИО в ММД представляют строки байтов с переменной длиной, но не больше 32 767 байтов. Каждая область может иметь до 32 767 ИО, а один архив - до 32 767 областей. Поскольку архивы размещаются в отдельные мультисписковые файлы, число архивов в ИБ связано с наличной генерацией операционной системы.

Отличительная особенность ММД является возможность реализации ИБ в неомогенной среде хранения. Среда хранения может включать как магнитные запоминающие средства, так и такие средства как терминалы. Оптические аудиовизуальные средства, управляемые ЭВМ и средства видеоэлектроники предусматриваются для включения в среде хранения на следующих этапах реализации ММД.

Для установления полного соответствия между координатами ИО в ИБ и ихнем физическим размещением в среде хранения, часть из ИО забронированы /рис.5

и рис.6./:

- для пользователя доступны все точки, имеющие положительные, ненулевые целые координаты;
- остальные точки используются для служебных целей. В частности:
 - в каждой области ИО с нулевой первой координатой содержат характеристики и физические адреса ИО области - это т.н. "основной индекс" области;
 - в каждом архиве "нулевая" область, т.е. все ИО с нулевой второй координатой, содержит характеристики областей архива и физические адреса их основных индексов - это т.н. "главная область" архива;
 - в ИБ "нулевой" архив, т.е. ИО с нулевой третьей координатой, содержит характеристики и местонахождение главных областей архивов ИБ - это т.н. "главный архив".

Отметим, что нумерация координат делается с право на лево.

ММД обеспечивает три основные операции с ИО:

- чтение;
- запись;
- перенесение из одной точки ИБ в другую, если это возможно в неомогенной среде хранения.

Операции чтение и запись можно проводить как с ИО в целом, так и с отдельными порциями ИО, длина которых произвольная, определяемая динамически пользователем.

Следует учесть, что физические характеристики среды хранения очевидно влияют на операции ММД.

3.3. СИСТЕМА ИНФОРМАЦИОННОГО КОНТАКТА /СИК/

СИК является инструментальным средством для создания конкретных САОИБ. СИК состоит из двух компонентов:

- ядро;
- оболочка.

Ядро СИК содержит:

- программы ММД;
- программы, выполняющие основные операции МИМ с ИО и областями. Операции с архивами и ИБ частично осуществлены в операционной системе.

В рамках оболочки вошли программы обеспечивающие:

- работоспособность системы;
- связь с операционной системой;
- диалог с пользователем;
- удобства для манипулирования с ИО - экранный редактор с встроенной возможностью для осуществления операций композиции и декомпозиции экранных ИО; средства наблюдения ИО и др.;
- защита от неправомерного использования системы и др.

В СИК возможно включать дополнительные программы, реализующие нестандартные операции с ИО и областями. Это, "пользовательское", развитие СИК названо "специализацией". Все программы ядра СИК открыты для использования при специализации. Программы специализации могут быть написаны на любом наличном в операционной системе языке программирования.

СИК способна работать с командными процедурами. Создание набора командных процедур для конкретного

пользователя названо "индивидуализацией".

На рис.8. дана принципиальная схема оперирования с СИК. Используя возможности СИК пользователь может например:

- создавать ИО, вводя их с клавиатуры терминала с помощью гибкого экранного редактора, пользуясь при необходимости услугами аппарата для предварительной композиции. В частности это может быть просто заполнение некоторых заранее подготовленных макетов;
- вводить в ИБ так созданный ИО в экранном и/или декомпозированном виде;
- послать ИО на другой терминал или на АЦПУ;
- наблюдать ИО из ИБ в нескольких режимах - поотдельности, по областям, в форме видеофильма, следуя некоторого индекса и др.;
- выполнять специфические пользовательские программы, включенные в СИК на этапе специализации;
- использовать индивидуальные командные процедуры и протоколы для ИБ.

На базе СИК с помощью средств операционной системы возможно построить многопользовательскую САОИБ, допускающую отдаленный доступ к ИБ /рис.9./.

3.4. МЕТОДОЛОГИЯ РАБОТЫ С СИК

Для облегчения осуществления развития и адаптации СИК разработан набор рекомендаций, формирующий единую методологию работы с системой, а в оболочке СИК включены соответствующие средства и возможности, учитывающие эти рекомендации. Таким обра-

ных САОИВ на базе СИК.

При разработки методологии взята в виду стратегия эволюционного синтеза систем, основанная на функциональном-структурном подходе к исследованию закономерностей развития и создания проблемно-ориентированных систем [5].

3.5. ОПЕРАЦИОННАЯ СРЕДА ФУНКЦИОНИРОВАНИЯ /ОСФ/

ОСФ состоит из набора технических и программных средств, такие как вычислительная система, различные аудиовизуальные аппараты и средства общения, операционные системы, пользовательские программы, реализующие разнообразные и специфичные для конкретных пользователей информационные процессы.

СИК оформлена в соответствии с операционной системой ДОС-РВ-Б, следует ее стандарту и не требует ограничений в операционной системе.

Минимальная конфигурация ОСФ для настоящей реализации СИК на базе СМ-4 ЭВМ включает:

- центральный процессор СМ 2104 или совместимые с ним;
- оперативная память с объемом 64 К слов;
- накопитель на кассетных магнитных дисках;
- терминал СМ 1604 - не меньше одного;
- накопитель на магнитной ленте;
- операционная система ДОС-РВ-Б или совместимые с ней.

Дополнительную периферию /печатающие устройства, более емкие диски, устройства управления архивами микрофильмов или видеотек/ можно добавлять.

4. ЗАКЛЮЧЕНИЕ

Рассмотренная в настоящей работе САОИБ была внедрена в одном из крупнейших болгарских химических комбинатов в городе Димитровград. На ее основе создана автоматизированная система диспетчерского управления комбинатом. Первый этап этой системы вошел в эксплуатацию.

САОИБ имеет широкое поле для дальнейшего развития. Возможно например:

- создание программного аппарата обслуживания на базе специализированного для ИВ языка высокого уровня;

- осуществление разных специализаций для оперирования с более крупными структурами /напр. реляционная, йерархическая, сетевая и др./, которые легко сводятся до основных структур МИМ используя специфику его индексации.

Автор благодарит проф. П.Барнев за поддержки при осуществлении всего проекта и реализации САОИБ, н.с. Ст.Порязов и асп. Б.Кокинов за особенно ценных разговоров в связи с этой работой, как и всех своих коллег, принявших участия в обсуждении и внедрении САОИБ.

ЛИТЕРАТУРА

1. P.Barnev. Systems for Information Servicing of Collectivities. SERDICA, vol.4, fasc. 2-3, SOFIA, 1978. 164-179.
2. П.Барнев, Кр.Марков. Проблемы управления в системах информационного обслуживания коллективов.

Сб. научно-исследовательских работ РГ-11/КНВВТ, том I.

TANULMANYOK 131, BUDAPEST, 1982. 15-23.

3. Kr. Markov. A Multi-domain Access Method. Coll. Research Papers of International Conference on Computer-based Scientific Research. PLOVDIV, Oct. 15-20, 1984.

4. Kr. Markov. A Microsystem for Information Contact. Proc. of the Fourteenth Spring Conference of the Union of Bulgarian Mathematicians. Sunny Beach, April, 1985. 471-475.

5. Е. Балашов. Эволюционный синтез систем. МОСКВА, Радио и связь, 1985.

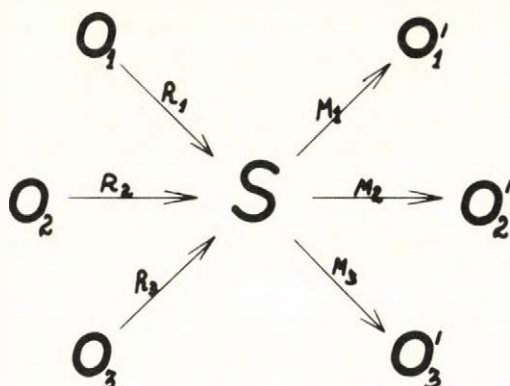


Рис.1.



Рис.2.

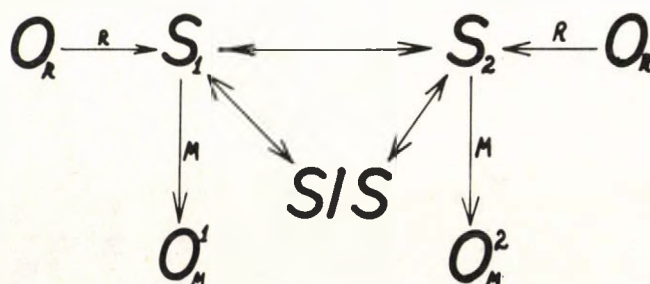


Рис.3.

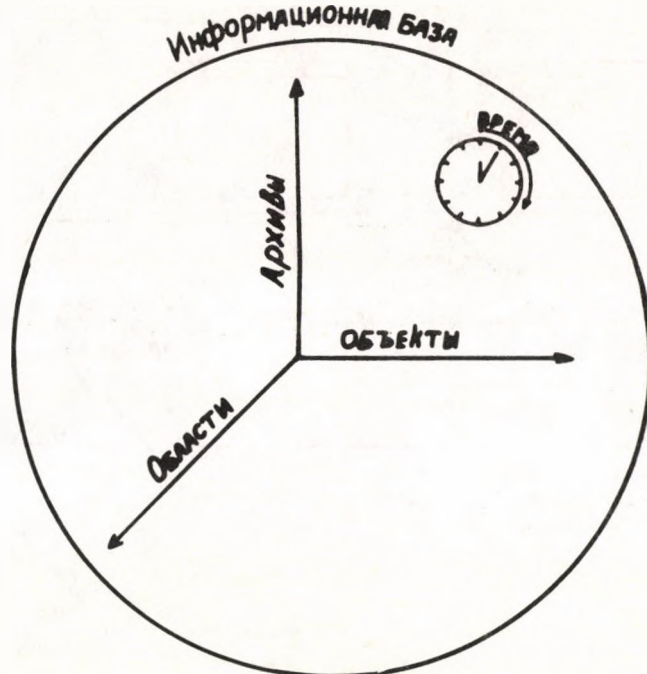


Рис. 4.

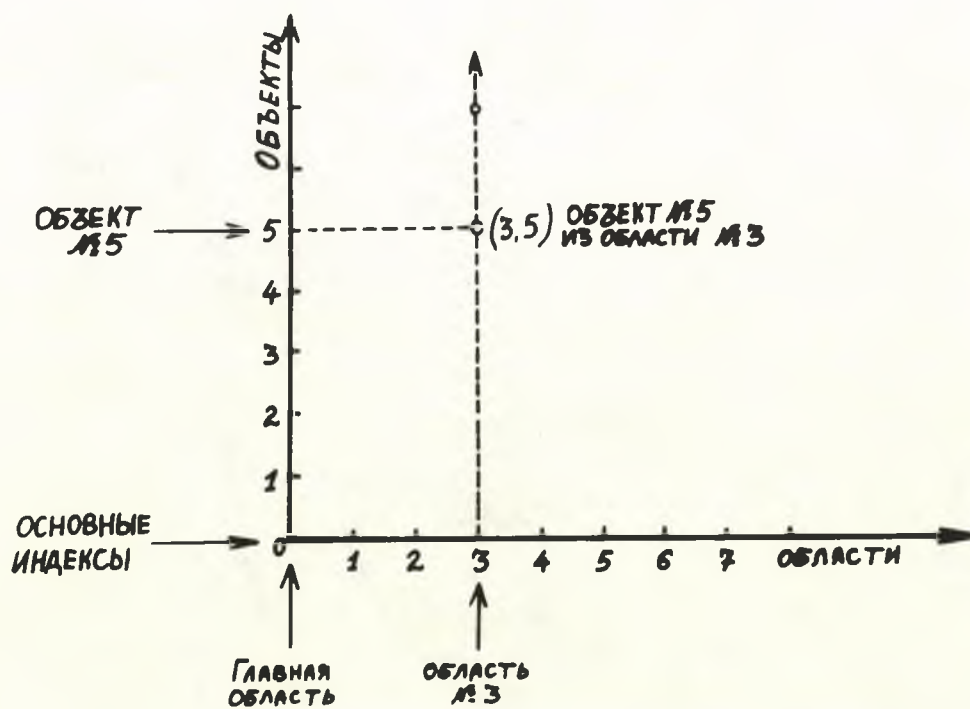


Рис. 5.

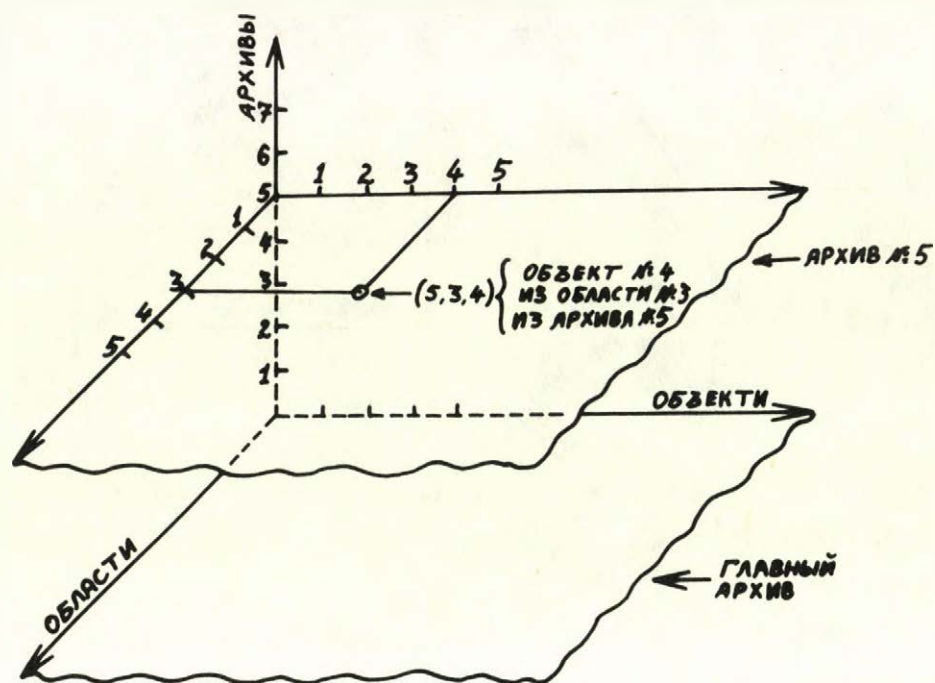


Рис.6.

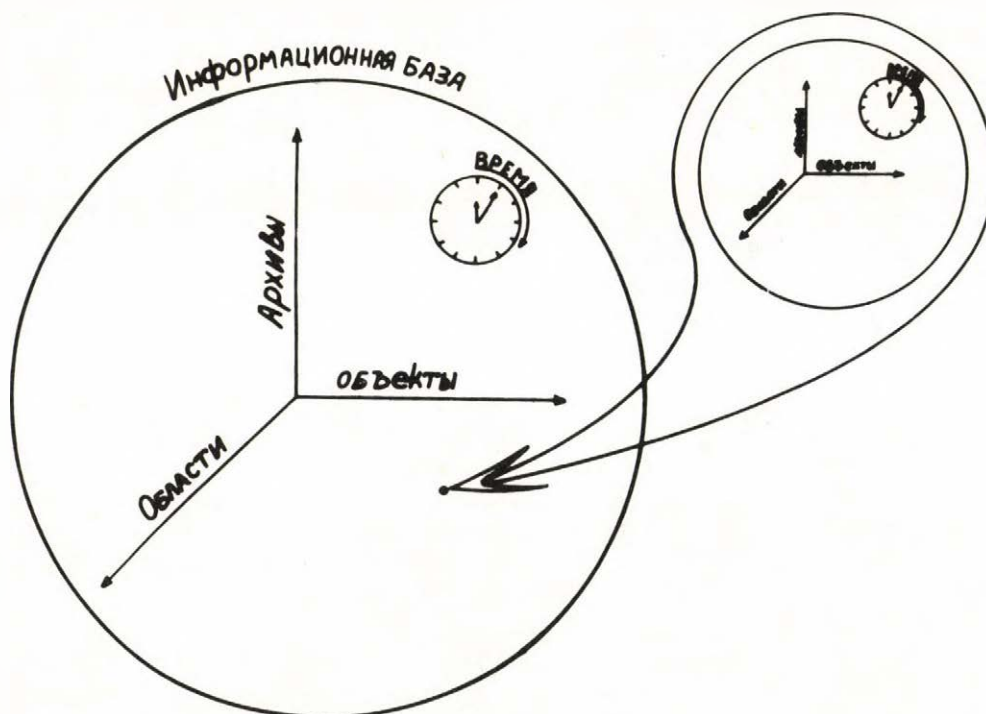


Рис.7

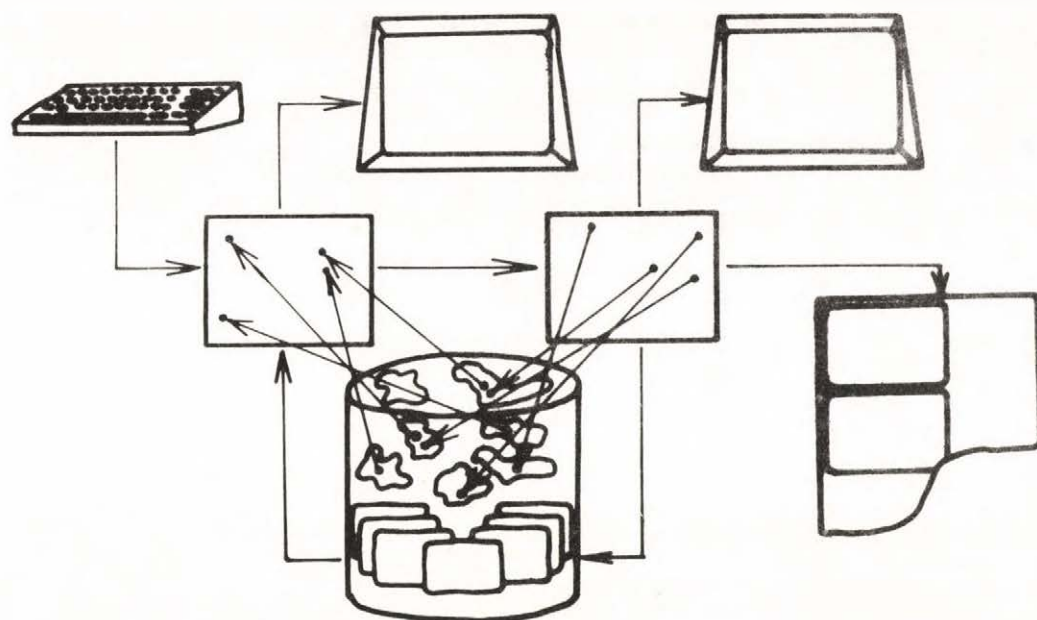


Рис.8

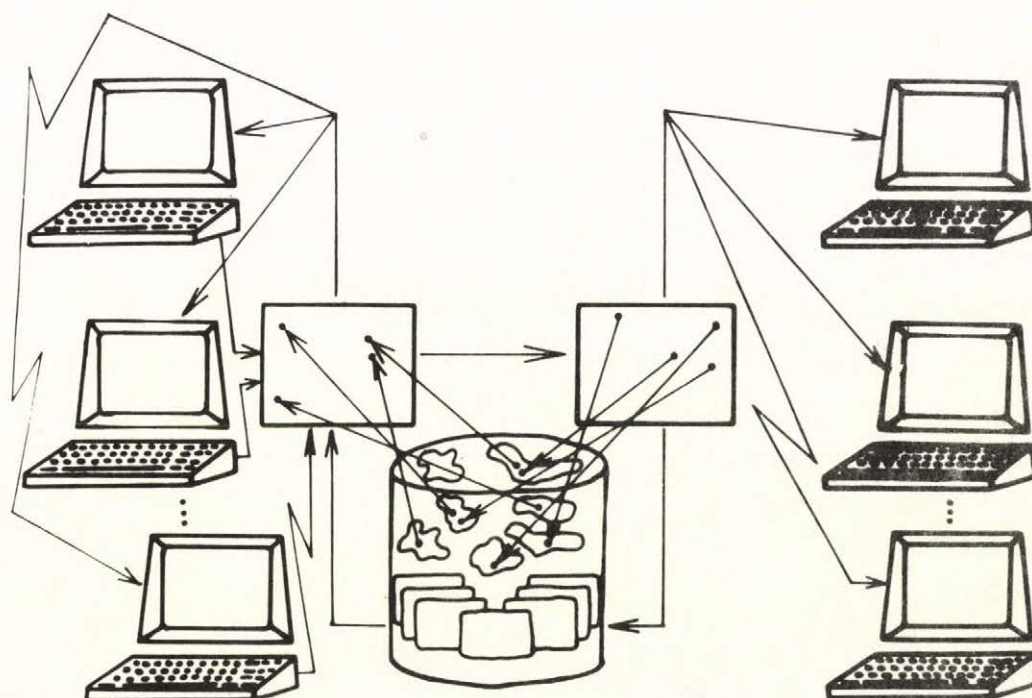


Рис.9.

СИСТЕМА АВТОМАТИЗИРОВАННОГО СОСТАВЛЕНИЯ СВОБОДНЫХ ТЕКСТОВ

ДО ВЬЕТ НГА

Институт ИК Ханой Вьетнам

Институт ММ с ВЦ БАН Болгария

Вопрос автоматизации работы с документацией с помощью компьютеров является одним из главных направлений применения вычислительной техники и автоматизации административной деятельности. Для этой цели был создан ряд текстообработывающих систем, при этом проблема автоматизированного синтеза документов разрешается двумя путями: или созданием универсальных систем, возможных составить всяческие виды документов или созданием специализированных систем для конкретного применения.

Первый подход отличается своей универсальностью, но именно эта универсальность задерживает скорость обработки конкретного документа. Второй подход, наоборот, ускоряет процесс создания конкретного документа, но может быть применим только для очень узкого круга применений.

Поставленный в данной работе подход является попыткой в поиске компромиссного разрешения вопроса. Разработанные модели системы на основании данного подхода отличаются достаточно повышенной скоростью и достаточно большим кругом применений. Системы организованы по следующей структуре: Каждая специализированная система /для конкретного применения/

состоится из базовой программной части и специфичной части для данного применения. Первая часть служит основой системы и создается программистом, а вторая часть легче создается и выполняется менее классификационными техническими специалистами. В целом система достигает универсальности за счет расширения специфичной части.

Первый метод, основанный на описанных концепциях, предлагает, что документы могут быть составлены из отдельных предварительно подготовленных фрагментов, в которых могут быть содержаны пустые поля, допускающие заполнение различными текстами по различным способам. Специфичная часть создается через так называемые макеты, которые содержат неизменяемые тексты фрагментов, указания пустых полей и возможные последовательности появления фрагментов в документах. Подробное описание этого метода было опубликовано в работе, совместно разработанной с Проф. П. Барневым /1/.

Второй метод предлагает, что для документов, относящихся к определенному классу, характерно относительно повторное использование определенных слов и словосочетаний. Это обстоятельство используется для сокращенного ввода слов и словосочетаний через их начальные буквы или через общепринятые/ для определенного круга пользователей/ сокращения.

Специфичная часть в этом методе получается путем создания специализированных словарей слов, часто использованных в данном виде документов.

Работа была выполнена в Лаборатории по при-

менении математики /г.Пловдив/ Института математики с ВЦ Болгарской АН. Автор выражает свою искреннюю благодарность Проф. Барневу за научную помощь, которую ей оказал в разработке данной работы, а также в целом периоде пребывания автора в Болгарии.

1. Общее описание метода.

Проблема механизации процесса выражения человеческого мышления или создания компьютерного инструмента для творческой работы интересует многих специалистов в области информатики-бизнеса. Разнообразие и нестандартность человеческого мышления делает автоматизацию выражения этого процесса очень трудным. Предлагаемый в данной работе метод основан на том, что всякие разработки выражаются текстовыми предложениями, которые можно разделить на **атомные единицы информации** - слова и словосочетания. Последние можно заранее подготовить и внести в ЭВМ для использования. Основная идея метода следующая:

В обыкновенном редакторе, удобном для исправления и ввода текста включается средство для ускорения ввода текста с помощью так называемого словаря. Словарь является списком наиболее часто использованных в данной работе слов, подготовленных самим составителем документа. В процессе ввода и исправления текста составитель в место ввода целого слова или словосочетания, имеющегося в списке, вводит только первые их буквы. Например, в словаре имеется словосочетание :

"институт математики с вычислительным центром"

тогда во время ввода текста в место этого длинного сочетания составитель вводит только группу знаков:

"и §"

где "§" — знак использования словаря, а "и" — первая буква словосочетания.

В словарь вводятся не все использованные слова, а только те, которые часто встречаются в тексте. Метод будет особенно эффективным если слова словаря различаются первой буквой, в этом случае имеется однозначное соответствие между словами и их первыми символами и вызов слов может быть реализован только через одну букву.

Описанный метод можно сравнивать с использованием сокращений в повседневном тексте. Например, часто встречается сочетание "и т. д.", что означает "и так далее", где "т" и "д" являются первыми буквами слов, а "." играет роль знака "§", т.е. указывает, что в данном случае идет сокращение.

Идея метода имеет аналогию в стенографии, где каждый слог обозначается своим обозначением. Более того, если научиться использовать стенографию означало бы тоже самое, что научиться новый язык, то при использовании описанного метода для ввода текста пользователи могут употреблять свои привычные обозначения.

Метод может быть также немного модифицирован путем реализации вызова слов из словаря не через первую группу их символов, а через удобные для пользователей обозначения. Например, слова "между" многие пишут "м/у", а словосочетание "научный сотрудник"

ник" пингует "н.с.". Если в методе понимаем под кодом сокращения группу первых символов слова то можем выражать принципиальную идею метода в реализации вызова слова через их коды. Если теперь в качестве кода будем принимать любой набор символов, объявляемый составителем кодом сокращения слова при вводе его в словарь, то получим модификацию метода. При этом, пользователь может использовать любые обозначения для своих сокращений.

Можно использовать также смешанный подход: вызов слов, имеющих специальный код-сокращения, реализуется через последние, а вызов остальных реализуется через группу их первых символов. Таким образом, пользователь должен ввести в словарь только свои специальные обозначения для некоторых слов, а те, для чего он не объявляет коды-сокращения будут иметь группу первых символов в качестве кода по умолчанию.

2. Проектирование метода. Редактор со словарем.

В данной части работы дается подробное описание метода при реализации.

В "редакторе со словарем" использование словаря допускается везде, где требуется ввод текста с клавиатуры и объявляется через знак сокращения "&". Ввод знака "&" вызывает следующие действия: выделение группы первых букв сокращенного слова, нахождение полного слова в словаре и наконец, записывание этого слова в обрабатываемый текст.

При выделении сокращения система ищет группу

знаков, стоящих между последним пробелом и символом "§". Например, при введенной строке:

"институт математики и ме§"

система выделяет группу "ме" в качестве сокращения. В этом анализе важно отметить, что последний пробел играет роль разделителя, а знак "§" играет роль и знака сокращения и разделителя.

При поиске полного слова в словаре, в системе предусмотрены 3 случая:

- если в словаре имеется только одно слово, начинающееся с введенной группы сокращения, то слово вводится прямо в текст

- если имеются несколько слов, начинающиеся с этой группы, то выводится на экране подсписок этих слов и предоставляется пользователю возможность выбора

- если же в словаре нет такого слова, то система вернет курсор экрана к положению знака сокращения "§" и ожидает окончания с клавиатуры. При этом предоставляется и возможность ввода данного слова в словарь.

Словарь возможных сокращений слов можно заранее подготовить и ввести в систему через редактор. Предусмотрена и возможность ввода слова в словарь в процессе ввода текста. Таким образом, даже на ходу работы пользователь может "договориться" с системой о своем сокращении. Эта возможность реализуется следующим образом: Например, при вводе строки "институт математики и механики", пользователь желает ввести слово "механики" в словарь. Для

этого он должен ввести знак "!" после желаемого слова, т.е. получается строка

"институт математики и механики!"

Если пользователь желает ввести в словарь словосочетание "математики и механики", то он должен ввести группу "!3". Система понимает при этом, что он хочет ввести в словарь последно введенное словосочетание, состоящее из 3 слов.

Для модификации метода - словарь сокращения с кодами, в словаре необходимо хранить кроме полных словосочетаний и их сокращений, а при вводе словосочетания в словарь в процессе работы с текстом необходимо объявить и его сокращения. Например, если пользователь хочет сообщить системе, что будет использовать сокращение "ММ" для вызова группы "математики и механики", то при вводе данного сочетания он должен ввести также группу "ММ3", т.е. вышевведенная строка получит вид:

"институт математики и механикиММ3"

и система будет понимать, что пользователь желает ввести сочетание, состоящее из 3-х слов с кодом сокращения "ММ" в словарь.

Во внутреннем представлении слова могут быть организованы по структуре дерева. Такая организация дает возможность экономии памяти ЭВМ и быстрого доступа к желаемому слову. Для примера рассмотрим список следующих слов: "желание, жених, журналист, компромисс, комплекс". Внутреннее представление данного списка описывается графом, дающимся на рис. 1.

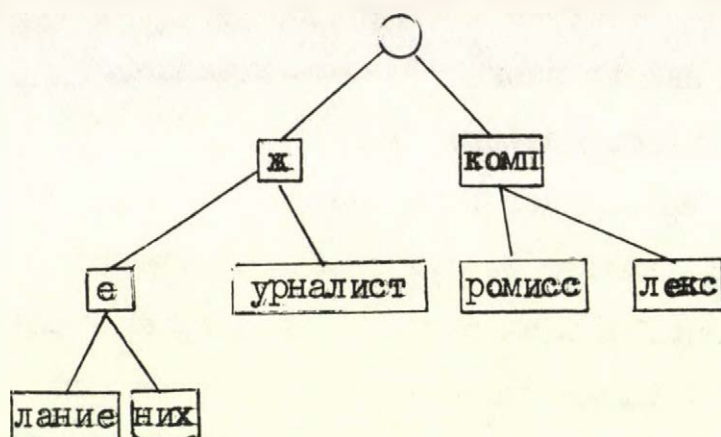


Рис. 1. Дерево структуры примерного списка.

Процесс анализа метода показывает, что он зависит от многих обстоятельств: от конкретного типа текста, от особенности языка, от конкретного человека, который непосредственно будет работать с системой, от компьютера, где будет реализована система и от др. Поэтому, для усовершенствования метода требуется время для накопления опыта и нахождения подходящих подходов.

Литература:

1. Бърнев П., Нга. Д. Система за автоматизирано съставяне на документи. Математика и математическо образование. София 1986, издателство БАН.

КОМПИЛЯЦИЯ ЗНАНИЙ В ЭКСПЕРТНОЙ СИСТЕМЕ ЭСАК

Стефан Сахно

Институт математики с вычислительным центром
Болгарская Академия Наук
София, Болгария

Экспертная система ЭСАК предназначена для частичной автоматизации /при помощи персональной ЭВМ/ обработки данных для археологических керамических находок.

Основной задачей является распознавание типа конкретного керамического изделия и его классификация в уже изученному керамическому комплексу.

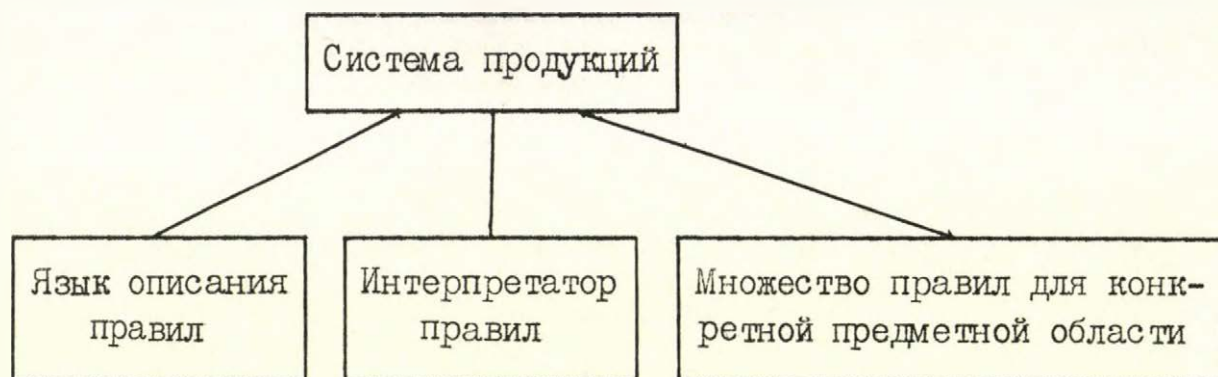
Рассматриваются общая функциональная структура экспертной системы и функции ее компонентов. Более подробно описываются метод представления знаний, подсистема поддержания базы знаний и процесс компиляции знаний.

1. Общая постановка задачи и подхода к ее решению.

Экспертная система распознавания и классификации археологической керамики ЭСАК представляет собой простую дедуктивную систему. Задача, которая она решает, - сделать вывод о принадлежности /или непринадлежности/ данного керамического изделия к определенному керамическому комплексу. Это осуществляется посредством системы продукций /СП/ на основе описания керамического комплекса и данных о конкретном керамическом изделии [6,7].

Основными компонентами СП /Фиг.1/ являются язык описания правил, интерпретатор правил и множество правил, заданных согласно синтаксису языка описания правил и представляющих по существу базу археологических знаний о керамическом комплексе.

Одно из преимуществ принятого подхода - ясное отделение общих знаний о проблеме /продукционные правила/ от информации о текущей задаче /данные/ и от процедур применения общих знаний при решении задачи /интерпретатор правил/.



Фиг.1. Основные части системы продукции.

Это дает возможность постепенно усовершенствовать и расширять знания о рассматриваемой предметной области посредством улучшения старых и добавления новых правил. Существует также возможность использовать интерпретатор правил для решения задач из другой предметной области, в которой знания могут быть представлены подобным способом.

2. Функциональная структура ЭСАК.

Специфические функции, выполняемые отдельными частями Экспертной системы, обособляют в ней четыре подсистемы: подсистема для поддержания базы знаний, подсистема для поддержания базы данных, подсистема для логического вывода и подсистема для обработки потребительского языка /Фиг.2/.



Фиг.2. Функциональная структура ЭСАК.

Подсистема для поддержания БД выполняет функции, связанные с

расширением и модификации базы. БД имеет простую логическую структуру. Содержимое ее разделов представляет последовательности из описании объектов. Разработан удобный для потребителя язык описания данных. Подсистема для поддержания БД осуществляет автоматический перевод описания введенного объекта на языке внутреннего представления данных.

Основной деятельностью, осуществляемой подсистемой для логического вывода, является распознавание и классификация керамических изделий. На этом этапе решения задачи применяется множество правил, чтобы обобщить данные о конкретном керамическом изделии и, если возможно, классифицировать его к данному комплексу.

Реализованная в подсистеме для логического вывода управляющая стратегия имеет смешанный тип. Модули интерпретатора правил осуществляют дедуктивный и индуктивный анализ [7]. В общем случае обобщение осуществляется посредством восходящего анализа по дереву заключений. Каждый из более общих признаков, до которых достигается, рассматривается как гипотеза, проверка которой осуществляется посредством анализа в глубину. Целью этой проверки является определение комбинации значений свойств керамического изделия, подтверждающей гипотезу. Если с участием потребителя уже сформулирована гипотеза о принадлежности исследуемого керамического изделия к комплексу, о котором система "знает", то проверка этой гипотезы осуществляется посредством нисходящего анализа по дереву заключений.

По существу алгоритм интерпретатора включает в себя рекурсивную обработку списочных структур. Ввиду этого языком программной реализации был выбран ЛИСП [5]. Процедуры логического вывода и все подсистемы ЭСАК представляют собой множество функции, дефинированные на ЛИСПе.

Воспринят диалоговый режим работы с системой. Общение осуществляется при помощи клавиатуры из которой потребитель вводит данные о конкретной задаче. Он участвует в диалоге главным образом ответами, которые дает на показанные на дисплее вопросы, заданные машиной. Указанный способ работы освобождает потребителя от необходимости запоминать все свойства керамического изделия, которые, возможно, будут иметь значение при решении задачи, и все команды потребительского языка.

Средства общения соответствуют диалоговому стилю работы. Создан

удобный потребительский язык, в котором включены: операторы с упрощенной структурой, при помощи которых потребитель вводит данные; вопросы со стороны системы, касающиеся определенных свойств исследуемого керамического изделия или связанные с выбором соответствующего режима работы системы; команды, при помощи которых потребитель указывает дейности, которые должна выполнить система.

Команды потребительского языка могут быть группированы согласно своему функциональному предназначению следующим способом: команды для использования базы знаний - расширение и модификация базы; команды для использования базы данных - расширение и модификация базы; команды для классификации; команды для обработки данных и др. Предусматривается дальнейшее детализирование и развитие языка для работы с системой.

3. Подсистема для поддержания базы знаний.

3.1. Представление знаний.

Археологические знания о керамических комплексах и некоторые общие археологические знания представляются в виде продукционных правил типа ЕСЛИ <условие> ТО <следствие> .

Правила, описывающие керамический комплекс, составляются на основе классификатора типов керамических изделий, разработанного археологами при исследовании комплекса. Классификатор представляет собой описание основных типов изделий посредством комбинации значений наблюдаемых свойств /форма изделия, размеры, состав глины и пр./.

Для описания правил создан удобный для эксперта язык, синтаксис которого дается на Фиг.3.

Правила, записанные таким образом, применяются прежде всего для логического вывода, при котором делается заключение о типе керамического изделия и о его принадлежности к керамическому комплексу. Множество правил каждого керамического комплекса формирует древовидную структуру /дерево заключений/ вида И/ИЛИ. Листья дерева обычно являются дескрипторами, т.е. они задают значения свойств исследуемых керамических изделий и их элементов. Узлы верхнего уровня дефинируются при помощи правил.

Основная конструкция, названная в синтаксисе "дескриптор" /Фиг.3/, применяется при описании правил и при описании объектов в базе данных.

```

<правило> ::= <имя> : ЕСЛИ <условие> ТО <следствие>
<имя> ::= <идентификатор>
<условие> ::= <факт> I <выражение>
<следствие> ::= <факт> I <следствие> <факт>
<факт> ::= <предикат> I ( НЕТ <предикаты> )
<факт> ::= <дескриптор>
<выражение> ::= <конъюнкция> I <дизъюнкция>
<конъюнкция> ::= <операнд> И <операнд>
<конъюнкция> ::= <конъюнкция> И <операнд>
<дизъюнкция> ::= <операнд> ИЛИ <операнд>
<дизъюнкция> ::= <дизъюнкция> ИЛИ <операнд>
<предикат> ::= <идентификатор>
<предикаты> ::= <предикат> I <предикаты> <предикат>
<дескриптор> ::= ( <элемент> <признаки> )
<элемент> ::= <идентификатор>
<признаки> ::= ( <признак> ) ( <признак> )
<признаки> ::= <признаки> ( <признак> )
<признак> ::= <свойство> <значение>
<свойство> ::= <идентификатор>
<значение> ::= <стоимость> I НЕТ <стоимости>
<значение> ::= ИЛИ <стоимость> <стоимости>
<значение> ::= > <атом> I < <атом> I В <атом> <атом>
<стойность> ::= <атом> I <список>
<стойности> ::= <стоимость> I <стоимости> <стоимость>
<атом> ::= <число> I <идентификатор>
<дескриптор> ::= ( <элемент> <признак> )
<операнд> ::= <факт> I ( <выражение> )

```

Фиг.3. Синтаксис языка описания правил.

Дескриптор имеет следующий вид:

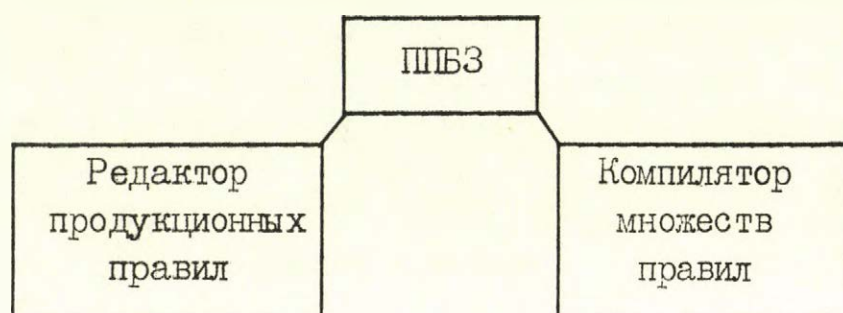
(<элемент> <свойство> <значение>) .

Содержание понятий "элемент", "свойство" и "значение" конкретизируется для каждой предметной области. В случае "элемент" это прежде всего часть керамического изделия, например шейка, тулово, рукоятка. "Свойство" это обозначение различных свойств элементов, например цвет, форма, положение, а "значение" - соответствующие значения этих свойств - желтый, двухконическая, вертикальная.

При помощи подсистемы для поддержания базы знаний /ШБЗ/ описанные таким способом продукционные правила вводятся в соответствующих разделах БЗ.

3.2. Функциональная схема ШБЗ.

Основными компонентами подсистемы для поддержания базы знаний /Фиг.4/ являются редактор продукционных правил и компилятор множеств правил.



Фиг.4. Основные компоненты ШБЗ.

Редактор - это средство предоставленное эксперту для ввода и изменения продукционных правил БЗ. При вводе автоматически осуществляется синтаксический анализ правил, т.е. распознаватель рассмотренной выше грамматики /Фиг.3/ анализирует каждое правило. Обнаруженные несоответствия с языком описания правил показываются вместе с необходимыми сообщениями на экране и эксперту дается возможность внести коррекцию.

Компилятор множеств правил осуществляет перевод введенного уже множества правил на языке внутреннего представления правил.

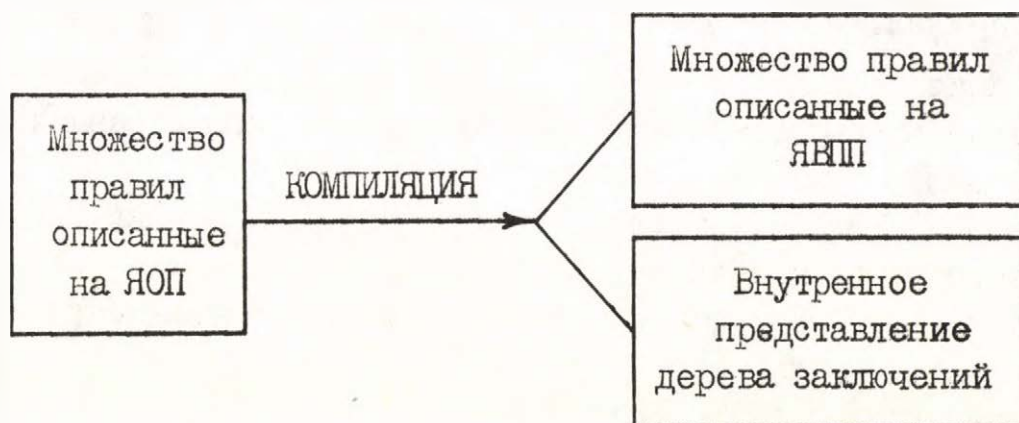
4. Компиляция знаний.

Предоставление эксперту удобного языка описания знаний и повышение быстродействия подсистемы для логического вывода /в основном - быстродействие интерпретатора/ являются две "противоречивые" цели при создании каждой экспертной системы. Если решим "усовершенствовать" интерпретатор с возможностями для обработки правил с более сложной структурой, то сразу получится понижение его быстродействие. Рассмотрим как решена эта проблема в экспертной системе ЭСАК.

4.1. Язык внутреннего представления правил.

Ввиду того, что во время использования системы потребителем очень

важно быстрое действие, интерпретатор правил создан для обработки правил с простой структурой. Он обрабатывает множества правил, записанные в БЗ на языке внутреннего представления правил /ЯВП/. Удобство языка описания правил ЯОП /Фиг.3/ заключается созданием и использованием компилятора для перевода множества правил с языка ЯОП на язык ЯВП /Фиг.5/.



Фиг.5. Компиляция множества правил.

Этот перевод осуществляется компилятором после ввода команды КОМПИЛЯЦИЯ экспертом и является "скрытым" для потребителя системы. Следующая компиляция рассматриваемого множества правил необходима только если будут сделаны изменения в нем. Синтаксис ЯВП дан на Фиг.6.

```

<правило> ::= ( <имя> ( <следствие> ) <условие> )
<имя>      ::= <идентификатор>
<следствие> ::= <факт> I <следствие> <факт>
<условие>  ::= <факт> I <операция> <факты>
<факт>     ::= <буква> <целое-без-знака>
<операция> ::= x I +
<факты>    ::= <факт> <факт> I <факты> <факт>
  
```

Фиг.6. Синтаксис языка внутреннего представления правил ЯВП.

Каждый факт в ЯОП /Фиг.3/ по существу представляет узел дерева заключений. Всем узлам компилятор дает служебные имена /на Фиг.6 <буква><целое-без-знака>/, которые как атомы ЛИСПа в процессе интерпретации несут в своих Р-списках информацию о фактах /в

смысле ЯОП/. Такое компактное представление правил способствует значительному уменьшению объема используемой оперативной памяти. Оно также оказалось очень удобным для анализа, которого выполняют "тяжелые" процедуры семантического анализа при вводе команды ЛОГИКА.

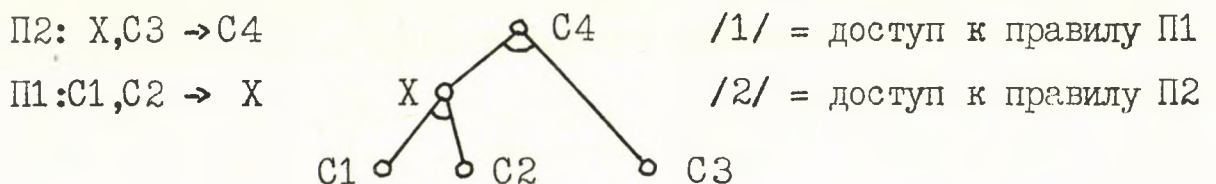
В системе ЭСАК компилятор выполняет еще одну важную функцию, связанную с дальнейшим повышением эффективности интерпретатора правил - связывание правил.

4.2. Связывание правил.

Быстродействие интерпретатора правил зависит от реализованных в нем методов для

- /1/ доступа к правилу, следствие которого содержит данный факт X /при нисходящем анализе по дереву заключений/, и
- /2/ доступа к правилу, условие которого содержит данный факт X /при восходящем анализе/.

Иллюстративный пример двух таких правил дан на Фиг.7.



Фиг.7. Пример двух правил, содержащих факт X.

Связывание продукционных правил множества М означает построение и записывание в базе знаний структуры, которой содержит информацию для обеспечения /1/ и /2/ для каждого факта X, участвующего в правилах М.

Как и всю компиляцию, связывание делается только один раз при вводе данного множества в БЗ. Все рабочие структуры /кроме множества правил на ЯВПП /Фиг.5//, которые создает компилятор /таблицы имен, таблицы "предшествования" и "наследования" правил и др./ названы в системе ЭСАК внутреннее представление дерева заключений /Фиг.5/.

5. Программная реализация.

Экспертная система ЭСАК разработана в секторе Математического обеспечения Института математики с Вычислительным центром БАН. Языком программирования был выбран ИКЛИСП [5]. Реализация системы сделана на персональной ЭВМ IBM PC XT. В своем настоящем варианте ЭСАК использует во время работы 256 К байтов оперативной памяти.

В заключении отметим, что воспринятый подход автоматизированного решения рассматриваемой археологической задачи, по нашему мнению и по мнению археологов, оказался перспективным. Результаты, полученные при экспериментировании системой с реальными керамическими комплексами Болгарии и Чехословакии, обнадеживающие и это нас импульсирует продолжить работу по ее развитию.

ЛИТЕРАТУРА

- [1] Уинстон, П., Хорн, Б., ЛИСП
/Эдисон-Уесли, 1981/ на англ. языке
- [2] Нильсон, Н., Принципы искусственного интеллекта
/Шпрингер-Ферлаг, 1982/ на англ. языке
- [3] Стефик, М. и др., Организация экспертных систем - руководство
/Искусственный интеллект, том 18, номер 2,
стр. 135-173, 1982 / на англ. языке
- [4] Хейес-Рот, Ф., Ватерман, Д., Ленат, Д./редакторы/,
Создание экспертных систем
/Эдисон-Уесли, 1983/ на англ. языке
- [5] ИКЛИСП - Руководство
/Интеграл-Куолити, Сиатл, США, 1983/ на англ. языке
- [6] Томов, В., Сахно, С.Т., Экспериментальная система
распознавания и классификации археологической керамики
/в сб. "Системы автоматизированного информационного
обслуживания", БАН, София, 1984/ на русском языке
- [7] Томов, В., Сахно, С.Т., Экспериментальная система
распознавания и классификации археологической керамики
/в Бибел, В., Петков, Б./редакторы/, Искусственный интеллект -
методология, системы, приложения,
Норт-Холланд, Амстердам, 1985/ на англ. языке

А.А.Стогний, Н.М.Глазунов

ПРОБЛЕМЫ ИНТЕГРАЦИИ В БАЗАХ ДАННЫХ

Объем знаний, накопленный человеческим обществом и слагаемый из индивидуальных знаний личностей, постоянно возрастает. Некоторые индивидуальные знания становятся всеобщими, а общечеловеческие знания, отраженные и преломленные отдельной личностью, дают ей фундамент для профессиональной деятельности и ориентиры в жизни. Однако рост совокупного объема знаний и усложнение общественных отношений приводят к тому, что человек теряет возможность получать нужные ему знания, а общество лишается ценнейших индивидуальных знаний.

Выход из подобной ситуации состоит в том, чтобы на основе индивидуальных знаний создавать интегрированную систему знаний, и наоборот, давать человеку необходимую ему сумму общечеловеческих знаний, обеспечивая взаимодействие между совокупными и индивидуальными знаниями.

Знания представляются в виде теорий и связаны с областями наук или практической деятельностью людей. Объективной основой интеграции знаний является, с одной стороны, материальное единство окружающего нас мира, с другой – процессы современной общественной жизни, характеризующиеся усилением функциональных взаимосвязей между всеми элементами общественной системы. Наука, продолжая оставаться системой знаний о мире, стала и непосредственной производительной силой, например, при создании сложных технических систем используются достижения не только областей производства, но и результаты многих наук.

Развитие современного научного знания обусловлено значительным числом факторов и процессов, среди которых определяющими яв-

ляются процессы дифференциации и интеграции, анализа и синтеза.

Дифференциация сужает область исследования, принося одновременно углубленное знание о ней. Интеграция как в частных науках, так и на основе междисциплинарных связей заключается в выявлении обобщающих закономерностей, в едином методологическом подходе к объектам исследования, возникновении интегративных наук и наук "стыковых", заполняющих "промежуточное пространство" между уже существующими и развитыми научными направлениями.

Развитие вычислительной техники, кибернетики и информатики и на их основе автоматизированных информационных систем (АИС) и систем баз данных (СБД) позволяет автоматизировать некоторые этапы процесса интеграции научного знания, в то же время конкретизируя и развивая их. Интеграция невозможна без предварительного анализа знаний и находится с ним в диалектическом единстве. Этап анализа при интеграции знаний в СБД включает информационный анализ предметной области (ПрО), её структуризацию и учет информационных потребностей пользователей. Важнейший вопрос, который возникает, заключается в том, чем обуславливается структуризация знаний и где её границы.

Автоматизированные информационные системы являются подклассом класса обобщенных динамических систем /1/, /2/. В настоящее время они в ряде случаев разрабатываются на основе систем баз данных. База данных, по Инглису, есть совокупность хранимых операционных данных, используемых прикладными системами некоторого предприятия /4/, /5/. Операционные данные отражают деятельность предприятия. Здесь под предприятием понимается функционирующий объект большого масштаба в области экономики, науки или техники. Если объект разбит на подобъекты, то каждому такому подобъекту может соответствовать своя база данных.

О базе знаний говорят, когда имеется одна или несколько баз данных, база процедур, элементы которой реализуют функции и отображения данной предметной области, а записи базы знаний содержат описания (в виде аксиом) объектов PrO и правила поведения или законы функционирования этих объектов. Причем в указанных описаниях используются переменные и функциональные символы, интерпретируемые в базе данных и базе процедур соответственно. Иногда в этом случае употребляют термин "база данных - база процедур - база знаний".

Выделяют различные классы АИС: информационно-поисковые, информационно-справочные, информационно-логические, информационно-распознающие /2/. Общим для них является то, что они представляют собой системы регистрации, переработки и хранения информации, предназначенные для поиска и выдачи необходимой информации в ответ на запросы или в режиме избирательного распространения.

Переход от эмпирического уровня объединения данных для решения классов задач к теоретическому осмысливанию информационных моделей предметных областей, возрастание роли теоретических исследований являются в настоящее время определяющими тенденциями развития банков данных (БД) и информационно-поисковых систем (ИИС). "Встраивание" средств вычислительной техники, методов кибернетики и информатики практически во все сферы человеческой деятельности, автоматизация на основе СБД самых различных процессов получают все большее распространение. В силу этого исследование логики и методологии развития систем баз данных получает сейчас общенаучный характер, далеко выходящий за рамки потребностей собственно данного научного направления.

Интеграция знаний в системах баз данных идет по следующим пяти основным направлениям: теоретический синтез знаний на мето-

дологическом уровне, математическая теория СБД, программное обеспечение СБД, интеграция по формам, представления и с использованием техники сетей связи и передачи данных. В последнее включается, в частности, интеграция локальных баз данных в информационно-вычислительных сетях. Объектом нашего рассмотрения будут, в основном, первые четыре направления интеграции знаний в СБД.

В настоящее время большое значение имеет фиксация основных результатов отдельного научного направления и классификация научных направлений. Согласно имеющимся данным, в мире сейчас насчитывается примерно 2000 областей знаний и каждая область делится еще на соответствующие подобласти. Ввиду этого актуальной является задача создания терминологических словарей науки, которые были бы толковыми не только для человека, но и для АИС. Необходимо разработать язык и систему представления результатов данного научного направления, структуру научной области и способы фиксации ее состояний.

Фиксация результатов научной работы в документальной форме (в виде статьи, обзора) позволяет описывать как результаты исследования, так и используемую методику и методологию. Сами же системы фиксации документальной информации в процессе проведения научного исследования носят и будут носить вспомогательный характер.

Из одной и той же работы разные читатели извлекают различную информацию. Поэтому стоит чрезвычайно сложная проблема объективного выделения и представления основных результатов публикации. Ряд подходов к решению этой проблемы имеется. Например, в японском проекте ЭВМ 5-го поколения в базах знаний будут фиксироваться не только результаты, но и методы их получения. Заметим, что решение этой проблемы не приведет к отказу от документального

описания проведенных исследований. По-видимому, по отношению к БЗ внешняя документальная база будет служить аналогом природы для естествоиспытателя. Процесс представления знаний в новой форме, т.е. в структурированном виде, потребует больших временных сроков.

На протяжении всего периода существования и развития науки получаемые в ней информация и знания фиксировались, в основном, следующими тремя способами:

1) в виде текстов, включающих иногда таблицы и изображения, в которых описывались задачи исследования, методики, результаты или излагалась некоторая теория. Этот вид информации, включающий монографии, статьи, отчеты, другие виды документов, относят к документальной;

2) в виде таблиц, схем, научных фактов, законов, кратко описывающих свойства или некоторые взаимосвязи исследуемых объектов, т.е. в виде структурированной информации, подмножеством которой является информация фактографическая;

3) в виде рисунков, карт, фотографий, голограмм и других видов изображений (образов) исследуемых объектов.

Отметим, что документальная информация может содержать текстовую информацию, структурированную и информацию в виде образов.

Условимся о некоторых понятиях. Если речь идет о представлении, пересылке и обработке текстов, фактов, понятий, таблиц, инструкций и т.п., представленных в буквенно-цифровой, оцифрованной, текстовой или графической форме безотносительно к тому, как и для чего они в дальнейшем будут использоваться, будем употреблять термин "данные". Если данные применяются для выбора возможных исходов из некоторого множества, то будем для обозначения этих определяющих исходы данных употреблять термин "информация".

Например, пусть у нас имеется программа обработки результатов статистических наблюдений. Тогда данные, полученные в результате статистических испытаний, будут одним из трех типов обрабатываемой программой информации наряду с промежуточными переменными и результатами (промежуточными и выходными). Цифровые данные, представляющие точки интерполяции функции одной переменной, являются входной информацией программы интерполирования.

Информацию, описывающую документ, называют документографической. В простейшем и в основном случае это есть просто библиографическая карточка описания документа. В настоящее время АИС для такого вида информации разработаны, и можно сказать, что проблема автоматизации документографических систем решена. Однако здесь остались (чрезвычайно важные) научно-организационные задачи. В то же время общая проблема автоматизации обработки, хранения и поиска документальной информации, несмотря на достигнутые здесь успехи, еще далека от своего решения.

При отображении информационных объектов в памяти ЭВМ возникает проблема отбора информации для хранения. Если для документальных систем в принципе ясно, какие объекты подлежат хранению, то для фактографической или приборной информации вопросы отбора и хранения стоят подчас чрезвычайно остро. Один метод отбора информации для хранения – задачный, заключается в том, что для описания объектов применяются и в дальнейшем хранятся те параметры, которые используются в задачах. Второй – информационный, состоит в разработке информационной модели исследуемого объекта и выделении хранимых параметров на основе этой модели. Здесь возникают проблемы оценки ценности информации, задачи ее старения.

Можно сказать, что системы баз данных возникли в связи с потребностями интеграции как по данным, так и по их обработке.

Проблема интеграции знаний, конечно, более общая, чем вопросы интеграции знаний в СБД. Развитие науки в направлении анализа, принося новые факты, информацию и знания, вместе с тем несет в себе опасность за огромным количеством частных об отдельных свойствах отдельных объектов утратить представление о целом. Цель интеграции знаний - преодолеть этот хаос частных, сформировать единый взгляд на предмет исследования с помощью не только самих знаний, но и методов их получения и синтеза.

Элементы этого процесса можно проследить на истории (довольно короткой) развития СБД. Эти системы, как и ИПС, принадлежат к классу автоматизированных систем обработки данных. В настоящее время выделяют три поколения таких систем. Системы первого поколения представляли собой наборы прикладных программ, решающих определенные классы информационно-вычислительных задач, причем каждая программа обрабатывала свои данные и наборы данных различных задач не интегрировались. Системы второго поколения разрабатывались на основе использования систем управления базами данных (СУБД), обеспечивающих интеграцию решаемых в системах задач по данным. В системах третьего поколения решаемые задачи интегрируются как по данным, так и по их обработке, что обеспечивается реализацией в этих системах развитого алгебраического (процедурного) или логического (непроцедурного) языка. Эта классификация применима для АСОД, которые интегрируют данные любых типов, однако в 70-х годах системы обычно интегрировали данные какой-либо одной формы представления, в основном текстовые или структурированные.

В настоящее время разработаны и разрабатываются СБД, интегрирующие текстовые и структурированные данные, структурированные и в виде образов, текстовые и речевые данные и другие комбинации

этих форм представления данных в СБД. Концептуальное направление интеграции СБД путем построения взаимных отображений различных моделей данных интегрирует сетевые, иерархические, реляционные и другие модели данных /7/. Возрастает интеллектуализация СБД. Одна из целей интеллектуализации – освобождение пользователя системы от забот, связанных с внутренним представлением данных. Использование формальных методов агрегации и дезагрегации накопленных данных также повышает интеллектуальность СБД. Методы агрегации данных уже представляются в руки пользователей СБД. Эти методы автоматизируют классификацию и кластеризацию и подсказывают человеку группировки в классы в таких предметных областях, где зачастую только интеграция базы данных и методов агрегации данных может привести к успеху. На основе развития языковых средств СБД развиваются также методы содержательной интеграции.

Общепризнанным в марксистско-ленинской философии является положение о том, что в современном познании материального мира одно из ведущих мест принадлежит математике, которая в то же время является и важнейшим средством интеграции науки. В разных науках процесс математизации идет различными путями, и проследить путь диалектического развития этого процесса в комплексах наук и в конкретной науке – задача, имеющая важное теоретическое и прикладное значение. Процесс математизации применительно к задачам интеграции знаний в СБД во многом аналогичен процессу математизации кибернетики, информатики и других наук кибернетического цикла. Он проходит в направлении использования моделей и методов теории множеств и комбинаторики, математической логики и теории алгоритмов, последующей алгебраизации с применением на всех этапах процесса математизации развитого аппарата математического анализа.

Так как СБД имеют дело с обработкой, хранением и поиском информации, то в изучении таких информационных процессов важная роль принадлежит методам теории информации, семиотики и математической лингвистики. На начальных этапах развития теории ИИС для описания моделей поиска документальной информации использовались теоретико-множественные модели, топологический аппарат теории толерантных пространств и методы математической статистики, а для описания моделей информационных систем – аппарат функциональных отношений, формальных грамматик и теории динамических систем.

В настоящее время в разработке СБД наиболее перспективным, по-видимому, является предложенный Коддом реляционный подход, основанный на использовании реляционных исчислений и реляционных алгебр. Заметим, однако, что при таком подходе учитывается семантика предметных областей, используется алгебраическая структура получаемых объектов (реляционных алгебр) и не учитываются статистические характеристики предметных областей. По-видимому, целесообразно объединить реляционный подход с развитыми в теории информации и кибернетике методами оценки информационных мер объектов. Аппарат математической логики и теории формальных систем используется в разработке концептуальных и инфологических моделей / 8 /. Интегрирующим и унифицирующим инструментом здесь, как и вообще в современной математике, является теория категорий. При интеграции в единое целое технических комплексов сетей передачи данных, их программного обеспечения и информационного наполнения единую методологию такой интеграции дает теория систем. Интеграция знаний – это такой вид представления знаний, который имеет отношение и к алгоритмам ее предварительной обработки и структуризации (например, представление в виде линейных файлов, графов и др.).

Интеграция знаний связана со способами представления знаний. Например, при представлении знаний с помощью логического исчисления интегрирующую функцию выполняет само исчисление, когда оно описывает заданную предметную область (мир пользователя) в виде набора аксиом и правил вывода. Аналогичную роль выполняют модели, основанные на семантических сетях, фреймах и др.

СПИСОК ЛИТЕРАТУРЫ

1. Глушков В.М. Основы безбумажной информатики. - М.: Наука, 1982. - 552 с.

2. Глушков В.М., Стогний А.А., Афанасьев Б.Н. Автоматизированные информационные системы. - М.: Знание, 1973. - 31 с.

3. Стогний А.А. Об основных принципах построения автоматизированных информационных систем. УСИМ, 1972, №2, с.3-II.

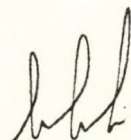
4. Информационные системы общего назначения/ Под ред. Е.Л.Ющенко - М.: Статистика, 1975. - 470 с.

5. Дейт К. Введение в системы баз данных. - М.: Наука, 1980. - 463 с.

6. Стогний А.А., Глазунов Н.М. О работе по системе СОАМИ. - В кн.: Вычисления в алгебре и комбинаторике. Киев: ИК АН УССР, 1978, с.41-53.

7. Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. - М.: Наука 1983. - 423 с.

8. *ANSI/X3/SPARC Study Group on Data Base Management System-Bulletin of ACM SIGMOD*, 1975, №2. - 409 p.



17.04.

РУКОВОДИТЕЛЬ В АВТОМАТИЗИРОВАННОМ УЧРЕЖДЕНИИ

Занг Конг Тхе ИМ с ВЦ БАН

Ул. Акад. Бончев, бк. 8, 1113 София, Болгария.

В работе рассматриваются основные деятельности руководителя и возможность их автоматизировать с помощью автоматизированных учрежденческих систем АУС.

I. ВВЕДЕНИЕ

Известно, что организация труда имеет большое значение в деятельности общества, при том организация труда руководящих кадров играет решительную роль в организации работы трудовых коллективов /3/.

Деятельность руководителя в современных условиях существенно усложняется. Требуется повышенное внимание ко многим вопросам, а следовательно, и больше времени на их решение. Основная часть личной работы руководителя связана с информационными деятельностями, сбором, сохранением, обработкой и распространением информации и специально с принятием решений. Руководители часто перегружены многими деятельностями, которые мешают им сосредоточить свое внимание на самые важные вопросы требующие творческую работу.

Поэтому распространение способов, позволяющих сократить трудоемкость различных операций в деятельности руководителей всех уровней имеет большое значение /1/.

Значительная часть этих деятельностей может быть существенно облегчена микро-ЭВМ или сетью микро-ЭВМ снабженными соответствующим математическим обеспечением /1,2/.

В данной работе обсуждаются основные деятельности руководителя, возможности автоматизации обслуживания этих деятельнос-

тей с помощью автоматизированных учреждений систем /АСУ/.

II. ОСНОВНЫЕ ДЕЯТЕЛЬНОСТИ РУКОВОДИТЕЛЯ

Работы руководителя разносторонние можно рассматривать несколько основных деятельностей, которые занимают значительную часть рабочего времени руководителя. Мы подробно останавливаемся на следующих главных видах такой деятельности /6/:

- принятие решений;
- проведение совещаний;
- работа с подчиненными;
- работа с документами;
- личные контакты: официальные и неофициальные встречи, командировки, принятие посетителей и проведение бесед;
- работа с техническими средствами для механизации и автоматизации управленческого труда;
- организация рабочего места;
- работа с секретарем.

По статистике в /9/, распределение времени административного работника высшего звена затрачиваемого на контакты в течение 8, 9 часов рабочего дня имеет следующий вид:

- составление писем	14%
- чтение почты	11%
- переговоры по телефону	16%
- неофициальные встречи	32%
- официальные встречи	15%
- командировки	6%
- анализ	3%
- производство	2%
- прочее	1%

И т о г о :	100%
-------------	------

Эти деятельности можно разделить на три вида:

- документы: составление писем, чтение почты;
- речь: переговоры по телефону;
- личные контакты: командировки, официальные и неофициальные встречи.

Из этой статистики видно, что около 95% рабочего времени руководитель расходует на обмен информацией, причем наибольшая часть времени приходится на деятельности связанные с обменом информацией, т.е. разговоры, встречи, документы. На принятие решений и решение проблем, т.е. фактически на выполнение основных обязанностей руководящих кадров, времени почти не остается /9/.

Эта ситуация одинакова для всех категорий руководителей. Процессы обмена информацией занимают большую долю рабочего времени у руководителя всех категорий. Поскольку у руководителей от 80 до 95% рабочего времени занимает передача, получение и обработка информации совершенствование процессов обмена информацией открывает огромные потенциальные возможности для регулирования общефирменных расходов, а также для повышения качества информации, исходя из которой руководящие органы принимают верные решения /7/. Анализ того, каким образом расходует руководитель это огромное время, раскрывает проблемы, процессы и характеристики обмена информацией у него.

III. НЕКОТОРЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ АВТОМАТИЗАЦИИ РАБОТ РУКОВОДИТЕЛЯ

В будущем автоматизированные системы, главная задача которых - совершенствовать выполнение только что упомянутых задач делового общения, обработки и обмена информацией, обеспечить возможность решения многих из названных здесь весьма актуальных задач. Эти задачи - повышение качества решений, производительности и более эффективно управление ресурсами ... Автоматизация всего цикла учрежденческих работ дает существенные выгоды, представляется логичным,

объединить многочисленные системы в виды обмена информацией. Развитие АУС, имеющих такую архитектуру, которая позволила бы учитывать требования конкретных пользователей и производить модульное наращивание, приведет к появлению рентабельных систем, решающих все проблемы обработки и обмена информацией /2/.

Многие средства, обеспечивающие приращение стоимости, выдают современную и актуальную информацию. Такая информация служит основой для принятия решений работниками руководящего звена и их действий. Информацию, от которой зависят действия такого рода, выбирают из различных источников как внутри, так и вне конкретной организации. Средства, предназначенные для связи и доступа к нужным видам информации являются важнейшими элементами большинства систем обеспечивающих приращение стоимости.

К системам приращения стоимости относятся системы связи, предназначенные для управляющих /например: речевая почта, электронная почта/, система поиска информации, система для подготовки решений и т.д. Эти системы способны дать лучшую и более своевременную информацию, заменить медленные методы /например, в почтовой связи/, обеспечить доступ к информации хранящейся в соответствующих базах данных или проведение анализа в целях принятия решений.

При рассмотрении основных деятельности руководителя было показано, что обмен информацией занимает большую часть рабочего времени, поэтому в данном разделе основное внимание уделим межличностной связи: разговоры, деловые встречи, обмен документами. Наконец, дается способ автоматизации планирования личного времени отводимого на конкретные виды деятельности руководителя.

3.1. Разговоры

Руководитель около 9% рабочего времени расходует на переговоры по телефону, а из них почти 50% случаев телефонных

вызовов он не может связать с нужными собеседниками. Даже в случае, если кого-то спрашивает, что-нибудь передать, большинство из которых не использует возможность передать полное сообщение /9/. Всего 9% телефонограмм представляет собой достаточно полное сообщение, позволяющие понять, что же на самом деле хотел абонент. Кроме того, телефонные вызовы часто прерывают ход мыслей и задерживают выполнение работы /9/.

Решить эти проблемы может система передачи речевых сигналов с накоплением и досылкой /НД/. Она дает возможность посылать и получать речевые сообщения. Возможны три вида НД: речевая почта, система предварительных вызовов, пользователь обращается к системе чтобы продиктовать или подготовить сообщение, применяя средства редактирования. Затем система запоминает сообщение и пересылает нужному абоненту. Получатель может сохранить, изъять сообщение, переслать его дальше или ответить на него, и можно передать то же сообщение нескольким адресатам.

Второй способ - электронная почта, одно из наиболее перспективных направлений автоматизации учрежденческой деятельности, задача которой - подготовка, распространение и обработка корреспонденции в рамках одного учреждения и с другими организациями /5/.

3.2. Деловые встречи

Для многих категорий служащих, особенно для управляющих и руководителей высшего звена деловые встречи - это единственный вид деятельности, который требует наибольших затрат времени. Они необходимы для того, чтобы ускорить процесс принятия решений и повысить их обоснованность. Деловые встречи охватывает процесс сбора, переработки информации и принятия решений. Имеются два вида деловых встреч /9/:

- коллективные встречи;

- индивидуальные встречи.

Для деловых встреч участникам приходится совершать поездки. Одна из главных причин - расширяется географическая сфера действия организации. Расходы на командировки большие /70% расходов приходится на такие встречи /9//. Исходя из этого замечания и результатов широких исследований и анализа больших объемов данных, фирма АТ&Т установила, что половина всех деловых встреч с разъездами может быть заменена конференц-связью. По определению в конференц-связи должны участвовать не менее трех человек, находящихся в разных географических пунктах, а сама конференц-связь осуществляется электронными средствами. В отношении организации и проведения конференц-связи пользователям должно обеспечиваться следующее /9/:

- Возможность резервирования оборудования для конференц-связи. Терминалы и средства доставки сообщения.
- Выбор порядка проведения конференции.
- Доступ к участию в конференции.
- Речевая связь /звук/.
- Возможность передавать и принимать видеосигналы, изображения и копии графиков, слайдов, диаграмм, текста ...

3.3. Документы

Обмен документами является предметом широких исследований, так как существует мнение о том, что улучшение этого может в значительной степени способствовать повышению как производительности, так и эффективности труда. В настоящее время существуют автоматизированные средства подготовки пересылки, приема, хранения, извлечения и контроля документов. Это весь цикл обмена документами. В числе важнейших рекомендованных мер можно отметить средства поиска информации, обработки речи и электронную почту. Фирма АТ&Т предложила систему электронного обмена документами, позво-

ляющую осуществлять сквозной обмен документами с помощью электронных средств. Говоря о полном обмене, имеют в виду весь цикл обмена документами, включая подготовку, пересылку, прием, хранение, извлечение и контроль.

3.4. Планирование времени

Чтобы работать эффективно большинству людей приходится заниматься планированием времени отводимого на конкретные виды деятельности. Руководитель действительно хорошо работающий, никогда не начинает с самой работы. Он начинает с планирования своей работы. Планирование личного времени занимает важное место в работе современных руководителей /4/. Таким образом, основным фактором повышения производительности труда руководителя являются средства, позволяющие руководителю более эффективно планировать свое время.

У каждого руководителя имеется индивидуальный распорядок дня, который находится на рабочем столе или носится при себе, даже в его памяти. По статистике в /4/, 80% руководителей не имеют постоянных графиков. Есть немало руководителей, которые не только не имеют постоянных графиков, но и не устанавливают заранее число и часы проведения совещаний. Таким образом доступ к информации ограничен и это затрудняет существенно координацию и организацию работ. Если бы распорядок дня, графики проведения совещаний и заседаний в организации, план и режим работы его заместителей хранились в оперативной памяти и были доступны другим служащим для целей планирования можно было бы высвободить время для более производительной работы.

Примером системы календарного планирования является СИСКАЛ, которая создается в ИМ с ВЦ БАН /1,6/. Система дает пользователю возможность назначать, отменять или переносить оперативные работы. Она следит за датой, временем начала и временем окончания работы, хранить нужную информацию о работе. В зависимости от заявок и манипуляции, система позволяет корректировать информацию о плановых работах сообщая

о наличии конфликтных ситуаций и предлагая решение, а также выдавать стандартные справки о расписании работ в течение года или определенного дня, годовой календарь и справки в зависимости от данного условия.

В первоначальном варианте система обеспечивает только индивидуальное планирование, однако конечной целью является обслуживание групповой деятельности. Ее значение возрастет если она будет использована в комплексе с другими автоматизированными системами как система кадров, система планирования и учета, система корреспонденций и решений
/1,6/.

IV. ЗАКЛЮЧЕНИЕ

В данной работе рассмотрены конкретные области, в которых АУС могут содействовать работе руководителей. Развивающиеся средства оргтехники предоставляют большие возможности для повышения производительности учрежденческого труда за счет внедрения в помощь сотрудникам учреждений специальных вычислительных средств. В будущем, руководитель будет прежде всего "многофункциональным работником". Он будет очень часто общаться и взаимодействовать с другими многофункциональными работниками с помощью множества средств /речевых, текста, видеоинформации/ как интерактивных, так и неинтерактивных. Он тоже будет выполнять задания, требующие большей степени творчества и больших знаний, чем теперь /2/. Техническая база будет продолжать расширяться и станет более производительной и доступной. Будут продолжать развиваться такие системы обмена информацией, как система передачи данных, электронные системы обмена документами, конференц-связь ... /8/. Одно лишь календарное планирование обеспечивает значительное повышение эффективности распределения времени для руководителей. Поскольку связь, необходимую для эффективного руководства, можно будет осуществлять с любого места, несомненно, изменятся приемы и методы работы руководителей, а также их качество труда. И в будущем руководителям при-

дется сталкиваться с очень сложными проблемами, однако, возможность эффективного решения этих проблем возрастет. В значительной мере в этом помогут системы обработки и обмена информацией. Располагая сетью гибких информационных систем руководитель будущего не только поднимает производительность труда, но и обеспечит значительно большие возможности для творческих достижений.

Л И Т Е Р А Т У Р А

- [1] Бърнев П., Бърнева М.: Автоматизиранна система за планиране на дейностите на ръководителя. Личен доклад.
- [2] Ершов А.П.: Автоматизация на работата на служещите. Персонални ЕВМ в задачите на информатиката. Новосибирск, 1984, с. 6-39.
- [3] Евгениев Г.: Социалистическа организация на труда на ръководителя. Партиздат, София, 1982.
- [4] Колцов В.И.: Организация на труда на хозяйственните ръководители. Москва, "Економика", 1979.
- [5] Тенев И.: Автоматизация на институционалните системи. София, "Техника", 1985.
- [6] Тхе Занг Конг: Автоматизирано планиране на работното време на ръководителя. Ще бъде публикуван в труда на конференцията на Съюза на българските математици, Слънчевият Бряг, 4.1986г.
- [7] King K.J.: Information Management Trends in Office Automation. Proceedings of the IEEE N.4, vol.71, 1983.
- [8] Lochovski F.H.: Improving office Productivity: a technology perspective. Proc. IEEE N.4, vol.71, 1983.
- [9] Teger S.L.: Factor impacting the evolution of office automation. Proc. IEEE, N 4, vol. 71, 1983.

РАСШИРЕНИЕ СИСТЕМЫ LISPKIT :
ИНТЕРПРЕТАТОР ФУНКЦИОНАЛЬНЫХ ПРОГРАММ
В ЛОГИЧЕСКОМ СТИЛЕ ПОСРЕДСТВОМ УНИФИКАЦИИ

Зоя Трифонова (ЕЦММ, Болгария)

Обсуждается интерпретатор функциональных программ, написанных на языке LISPKIT , в логическом стиле посредством унификации. Проводится параллель с другими известными попытками интегрировать логическое и функциональное программирование.

Язык LISPKIT (или инструментальный Лисп) – строго функциональный вариант Лиспа, описанный Хендерсоном [1] . Для экспериментирования в области функционального программирования используется собственная реализация системы LISPKIT при помощи абстрактной SECD –машины [2] . Реализация сделана на языке PL/1 для больших ЭВМ. Наш опыт с системой LISPKIT подтверждает тематические свойства функционального языка. При продолжении примитивных функций языка LISPKIT до их превращения в тотальные путем введения неопределенных значений, становится возможным применение математического аппарата, разработанного Манной [3] – больше всего теория наименьшей неподвижной точки функциональных программ. Нами доказана теорема о сведении некоторых рекурсивных программ к итеративным, а также и теорема о разложении некоторых рекурсивных программ в виде бесконечного условия.

Полученные результаты позволяют исследовать прог-

рамм при помощи математических методов и построение более интеллигентного компилятора.

Язык **LISPKIT** уже довольно популярный, очень часто цитируется в литературе, развивается в разных аспектах. Интересные расширения системы **LISPKIT** описаны в работах [4] и [5] .

В [4] показаны результаты в связи с расширением языка **LISPKIT** путем добавления новых примитивных функций. Используется способ **bootstrapping** , так как транслятор из **LISPKIT** -а в **SECD** - машинный код написан не только на языке абстрактной машины, но и на самом языке **LISPKIT** .

В [5] предложен **lazy** -вариант языка **LISPKIT** , который дополнен недетерминистическими конструкциями. Цель этого расширения сделать функциональный язык подходящим для описания и реализации вычислительных системах, у которых может быть сложное параллельное и асинхронное поведение. Виртуальная машина для аппликативного мультипрограммирования, которая использована и в этом случае промежуточным шагом, является вариантом **SECD** - машины.

Наш следующий шаг при экспериментировании с системой **LISPKIT** - написание интерпретатора функций, работающий в логическом стиле. Теоретическая обосновка возможности интерпретировать функциональные программы в логическом стиле, представлена в [6] Раденским. Эта идея развита в одной конкретной системе для функционального программирования - в системе **LISPKIT/L** . Настоящая работа посвящается

именно этой системе.

Исполнение функциональных программ в логическом стиле может рассматриваться как редукция функциональных выражений. Основные правила логического вывода, которые применяются в процессе редукций, являются определениями примитивных функций языка **LISPKIT**. Выбрана такая редукционная стратегия, при которой вычисление аргументов задерживается до тех пор пока это необходимо.

Интерпретатор языка **LISPKIT** в логическом стиле включает возможности выполнения программ с конкретными аргументами и с аргументами, в которых участвуют переменные определенных типов. Пользователь может задавать термины, в которых должен быть выражен результат. При обращении к функции с переменными в аргументах генерируются множества альтернативных решений - получается вид единственных возможных решений в зависимости от вида единственных возможных аргументов.

1. Описание языка

1.1. Формальное описание синтаксиса

определение ::=

функциональное_выражение = терм . |

условное_функциональное_выражение = терм.

цель ::= ? функциональное_выражение.

условное_функциональное_выражение ::=

функциональное_выражение [условие]

терм ::= целое | символный_атом | переменная |

функциональное_выражение | список

условие ::= терм отношение терм
 отношение ::= = | /=
 символьный_атом ::= прописная_буква |
 символьный_атом прописная_буква |
 символьный_атом цифра
 прописная_буква ::= * | - | A | B | C | ... | Z
 функциональное_выражение ::=
 символьный_атом (ряд_аргументов)
 ряд_аргументов ::= терм | ряд_аргументов , терм
 список ::= < > | < ряд_аргументов > | < терм . терм >
 переменная ::= x_переменная | a_переменная |
 m_переменная | k_переменная
 x_переменная ::=
 признак_произвольного_терма остаток
 a_переменная ::= признак_атома остаток
 m_переменная ::= признак_целого остаток
 k_переменная ::=
 признак_целого_неотрицательного остаток
 признак_произвольного_терма ::= o | p | ... | x | y | z
 признак_атома ::= a | b | ... | f | g | h
 признак_целого ::= l | m | n
 признак_целого_неотрицательного ::= i | j | k
 остаток ::= строчная_буква | цифра |
 остаток строчная_буква | остаток цифра

1.2. Неформальное описание синтаксиса

1.2.1. Определения. В системе можно определять функции как и в языке LISPKIT ([1] , абстрактная форма). Имеются и дополнительные возможности.

1.2.2. Цели. Цель - это заявка на выполнение функции с данными аргументами, в которых могут участвовать свободные переменные.

1.2.3. Переменные. В общем переменным в языке

LISPKIT/L могут сопоставляться термы. Имена переменных несут информацию о том, над какими множествами из термов могут варьировать переменные.

x _переменные варьируют над множеством всех термов.

a _переменные варьируют над множеством из символьных атомов, числовых атомов, a _переменных, m _переменных и функциональных выражений вида $s(\dots)$ и $-(\dots)$, которые означают атомами.

m _переменные варьируют над множеством числовых атомов, m _переменных, k _переменных и функциональных выражений вида $s(\dots)$ и $-(\dots)$.

k _переменные варьируют над множеством неотрицательных целых, k _переменных и функциональных выражений типа $s(\dots)$.

1.3. Примитивные функции

В системе встроены примитивные функции языка

LISPKIT следующим образом:

- ① $ATOM(a1)=T.$
- ② $ATOM(CONS(x1,x2))=F.$
- ③ $CAR(CONS(x1,x2))=x1.$
- ④ $CDR(CONS(x1,x2))=x2.$
- ⑤ $EQ(a1,a2)[a1==a2]=T.$
- ⑥ $EQ(a1,a2)[a1/=a2]=F.$

- ⑦ $EQ(a1, CONS(x1, x2)) = F.$
- ⑧ $EQ(CONS(x1, x2), a2) = F.$
- ⑨ $LEQ(m1, m2) = LEQ0(SUB(m1, m2)).$
- ⑩ $LEQ0(0) = T.$
- ⑪ $LEQ0(S(k1)) = F.$
- ⑫ $LEQ0(-(S(k1))) = T.$
- ⑬ $ADD(m1, 0) = m1.$
- ⑭ $ADD(m1, S(k2)) = ADD1(ADD(m1, k2)).$
- ⑮ $ADD(m1, -(S(k2))) = SUB(m1, S(k2)).$
- ⑯ $ADD1(0) = 1.$
- ⑰ $ADD1(S(k1)) = S(S(k1)).$
- ⑱ $ADD1(-(S(k1))) = -(k1).$
- ⑲ $SUB(m1, 0) = m1.$
- ⑳ $SUB(m1, -(S(k2))) = ADD(m1, S(k2)).$
- ㉑ $SUB(m1, S(k2)) = SUB1(SUB(m1, k2)).$
- ㉒ $SUB1(0) = -1.$
- ㉓ $SUB1(S(k1)) = k1.$
- ㉔ $SUB1(-(S(k1))) = -(S(S(k1))).$
- ㉕ $MUL(m1, 0) = 0.$
- ㉖ $MUL(m1, S(k2)) = ADD(m1, MUL(m1, k2)).$
- ㉗ $MUL(m1, -(S(k2))) = SUB(MUL(m1, -(k2)), m1).$
- ㉘ $REM(m1, m2) = *REM(m1, m2).$
- ㉙ $DIV(m1, m2) = *DIV(m1, m2).$
- ㉚ $IF(T, x1, x2) = x1.$
- ㉛ $IF(F, x1, x2) = x2.$

Предусмотрено использование функций *ADD ,
*MUL , *SUB и др. для более эффективного выполнения
арифметических действий.

2. Пример интерпретирования функции, написанной на языке LISPKIT/L , в логическом стиле.

2.1. База определений

Первоначальная база определений состоит из определений всех примитивных функций. Определения в памяти сохраняются в "скелетной" форме. В этой внутренней форме не сохраняются имена переменных. Каждое использование некоторого определения вызывает генерирование нового набора переменных, связанных с этим определением.

Правила вывода в системе – это именно определения функций из базы определений.

Будем считать, что правила вывода пронумерованы. Так удобнее объяснить пример. (Например второе использование правила (2) вызывает генерирование переменных $x1_{(2)_2}$ и $x2_{(2)_2}$.)

2.2. Редукция функциональных выражений

Обращение к функции с конкретными аргументами представляет функциональное выражение. Интерпретатор редуцирует это выражение до получения результата, соображенного с пользователем.

На каждом шагу редукции, некоторый подтерм сопоставляется левой части некоторого определения из базы определений (унифицируется) и заменяется на его правую часть. Подтерм, который заменяется – это самый левый подтерм, который может унифицироваться.

Использование такой редукции задерживает вычисление аргументов (с лево направо) до тех пор, пока это необходимо.

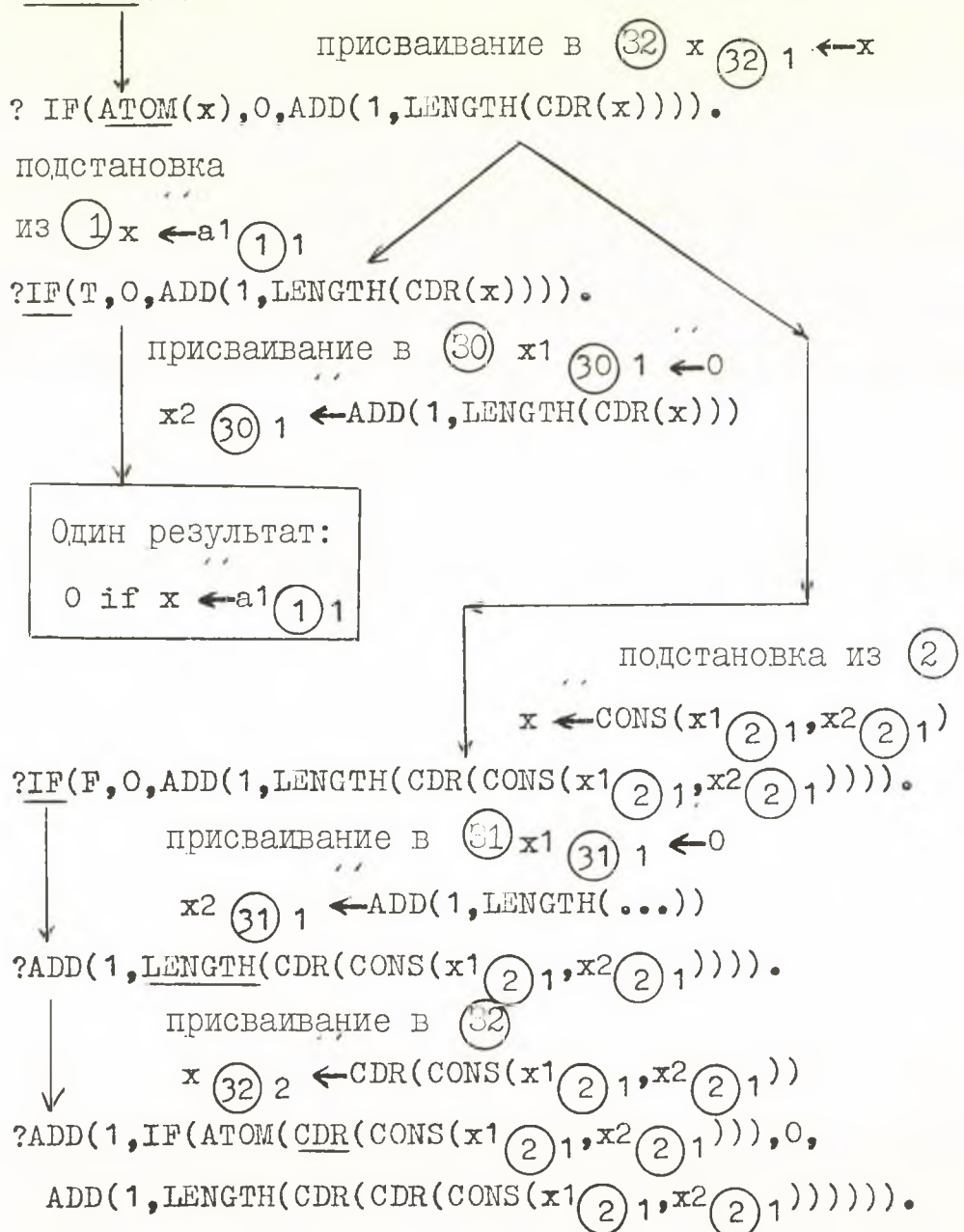
Процес унификации может включать присваивания переменных или подстановки переменных. Результат после одного шага редукции может быть списком нескольких альтернатив.

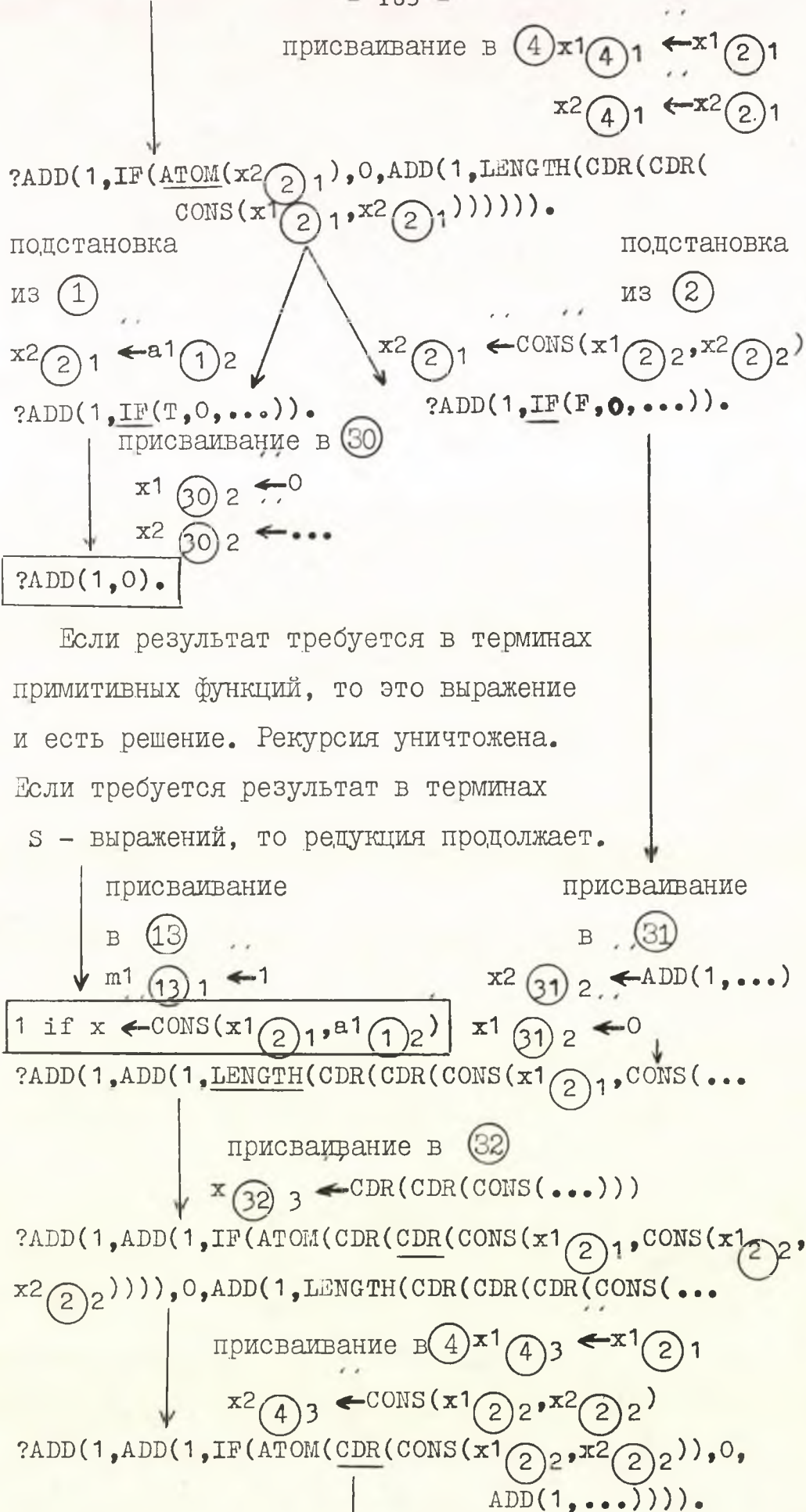
2.3. Функция LENGTH . Находит длину списка и определяется (в соответствии с описанным синтаксисом) следующим способом:

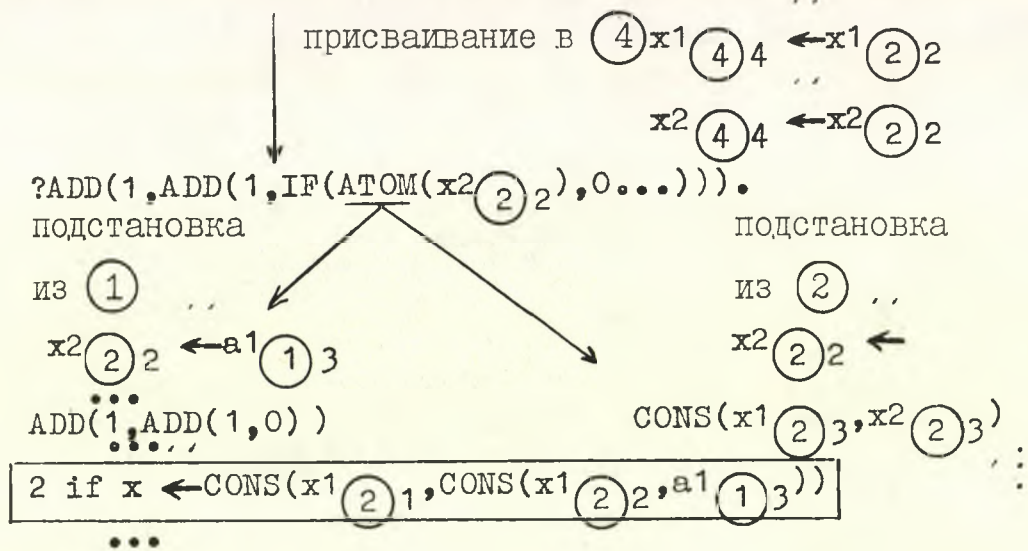
(32) LENGTH(x)=IF(ATOM(x),0,ADD(1,LENGTH(CDR(x)))).

Следует схема выполнения цели ?LENGTH(x)

?LENGTH(x).







Видно, что когда на вид решения не навязаны другие условия, получается бесконечный ряд ответов.

Пользователь, однако, получает более наглядный результат. Примерно такого типа:

0 if $x \leftarrow a1$
 1 if $x \leftarrow \langle x1.a1 \rangle$
 2 if $x \leftarrow \langle x1 x2.a1 \rangle$
 3 if $x \leftarrow \langle x1 x2 x3.a1 \rangle$
 ...

Т.е. длина атома 0, ..., длина списка из n элементов $-n$.

3. Экспериментальная реализация

Интерпретатор реализован на языке Паскаль для микрокомпьютеров IBM-PC

4. Краткий обзор других попыток интегрирования функционального и логического программирования.

4.1. Система LOGLISP [7]. Это LISP, расширенный функциями, обеспечивающими в частности эквивалент языка PROLOG. Авторы имеют богатый опыт в использовании языка LISP. Их цель - не выходя из знако-

мого LISP – окружения, располагать возможностями логического программирования.

Управляющая стратегия выбирается пользователем. В частности можно задать механизм **backtracking** и достичь возможности языка PROLOG

4.2. Система QLOG [8]. Является экспериментальной программной средой для PROLOG. Она достигнута посредством языка LISP и встроенная в язык LISP. Встроением PROLOG –а в LISP –е, оба языка взаимно обогащаются. LISP способствует это своим хорошо развитым программном окружением, а PROLOG – своими механизмами распознавания по образцу, методологией программирования, ориентированной к спецификациям и др. Система QLOG реализована на INTERLISP, FORTRAN LISP F3, STANFORD LISP 1.6, FRANZ LISP для VAX/780 и на LISP для LISP –машин.

4.3. Система PiL [9]. Это легкая реализация PROLOG –а написанная на языке LISP.

4.4. Система Qute [10]. Выражения в Qute – это или PROLOG-выражения, или LISP –выражения. PROLOG – часть и LISP – часть вызываются взаимно рекурсивно.

4.5. Система EQLOG [11]. Это система для логического программирования. Основана на полной формальной системе – наименьшая логика, которая имеет как клаузы Хорна, так и равенства. В терминах этой самой логики дана точная семантика тех средств, которыми логический язык дополнен: функциональное программирование, абстрактные типы данных и др.

4.6. Система FUNLOG [12] . Это вычислительная модель, интегрирующая логическое и функциональное программирование. Обеспечивает вычисления бесконечными структурами данных.

4.7. Система FPL [11,12] . Возможности системы как у функционального языка, а синтаксис – как у логического. Реализован механизм *lazy* – вычислений, имеется возможность манипулирования бесконечными структурами данных.

4.8. Система PROLOG/KR [11,12] . В системе некоторые образцы выполняются.

4.9. Система POPLOG [13] . Интегрирует языки LISP , POP-11 , и PROLOG . Процесс трансляции проходит через промежуточный шаг – трансляция до языка одной абстрактной машины.

4.10. Виртуальная L – машина для логического и функционального программирования [14] . Проект разработан в Софийском университете как часть среды для программирования на языках LOGO , PROLOG и LISP .

4.11. Обобщение

В литературе нашли отражение следующие принципиальные возможности объединения свойств функционального и логического программирования:

1/ Остаться в окружении логического языка, но добавить возможности эффективного вычисления функций.

В этом случае возможности системы для логического программирования обогащаются, а пользователь остается в знакомой программной среде, в которой

имеет традиции и опыт.

2/ Остаться в окружении функционального языка, но добавить функции, обеспечивающие эквивалент логического языка.

При такой реализации, к возможностям системы для функционального программирования, добавляются механизмы, которые используются в системах логического программирования или добавляется вся сила некоего логического языка. И в этом случае пользователь остается в знакомой программной среде.

3/ Создать многоязыковую среду программирования, в которой реализована возможность определять и выполнять как функциональные, так и логические программы.

Такая среда предоставляет средства решения большого класса реальных программистских задач, но требует усваивания двух стилей программирования.

5. Новый подход к объединению функционального и логического программирования.

В системе LISPKIT/L реализован оригинальный подход к объединению двух стилей - включен интерпретатор функциональных программ, работающий в логическом стиле. Нашли применение идеи логического вывода и логического программирования. Однако пользователь остается в окружении очень простого строго функционального языка.

1. Henderson P. "Functional Programming, Application and Implementation ", Prentice Hall, 1980
2. Radensky A., Trifonova Z. "A language for pure functional programming - description and implementation", Proc. of the Twelfth Spring Conference of the Union of Bulgarian Mathematicians, 1983
3. Manna Z. "Mathematical Theory of Computations", McGraw- Hill, 1974
4. Stojković V. i dr. "LispKit Lisp - jezik verzija ARL", Bilten za nauka i informatiku SAPV, 1984
5. Abramsky S., Sykes R. "Secd-m : a Virtual Machine for Applicative Multiprogramming", Research Report DoC 85/xx, August 1985
6. Radensky A. "Functional Programming in the style of logical programming", C.R. Bulgarian Academy of Sciences, 37, 6, 1984, 741-744
7. Robinson J., Siebert E. "Loglisp: motivation, design and implementation"- "Logic Programming", London, 1982, 299-313
8. Komorovski H. "Qlog- the programming environment for Prolog in Lisp"- "Logic Programming", London, 1982, 315 - 322
9. Wallace R. "An easy implementation of PiL (Prolog in Lisp)", SIGART Newsletter, 85, 1983, 29-31
10. Sato M., Sakurai T, "Qute: a Prolog/Lisp type language for logic programming", Proc. IJCAI-83, III, 3, 1983, 507-520
11. Goguen J., Meseguer J. "EQLOG: equality, types

"Functional and logic programming", 1983

12. Subrahmaniam P., You J, "FUNLOG = Functions plus Logic : a computational model integrating logic programming and functional programming", Technical Report UTEC - 83 - 040, 1983

13. Hardy S., Gibson J., Mellish C., Sloman A., Cunningham J. "POPLOG - a multipurpose multi - language program development environment", System Designers International PLC , 1985

14. Filimonov K., Georgiev G, " Virtual L- machine for functional and logic programming", Report, Bulgarian AI Seminar, 1985

СТРУКТУРИЗАЦИЯ БУЛЕВЫХ ДАННЫХ НА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ

Г. Хенчей

1. ВВЕДЕНИЕ

Использование вычислительной техники для структуризации многомерных данных с целью выявления сложных эмпирических зависимостей - важная область автоматизации научных исследований. До недавнего времени развитие программно-алгоритмического обеспечения этой области было ориентировано на большие машины. Аналогичные разработки для персональных машин сильно отстают.

Цель настоящего сообщения - показать, что создание соответствующего обеспечения для микротехники не только актуально, но и практически достижимо.

Мы ограничиваемся рассмотрением одной конкретной задачи - реализации метода аналога, т.е. выделения в большом массиве подмассива "похожих" на заданный объект.

В кругу задач структуризации данных это новая задача, хотя в практических разработках сложных информационных систем она неосознанно используется очень давно. Причина, почему была выбрана эта задача для рассмотрения, обусловлена следующими тремя обстоятельствами:

- а/ ее можно легко приспособить для решения всех основных традиционных задач структуризации: кластерного анализа, группировки признаков, выделения наиболее информативного подмножества регрессоров и т.п.;
- б/ в отличие от традиционных задач структуризации, являющихся в подавляющем числе "плохими" с точки зрения современной теории сложности, эта задача допускает эффективное решение, причем настолько эффективное, что можно использовать микромашины /персональные компьютеры/ для практических больших

массивов информации;

в/ будучи осознанной как самостоятельная новая задача структуризации, она выявляет новое направление приложений для методов обработки сложно организованных больших массивов информации.

Не последнее значение сыграло и то, что формальная постановка ее была предложена автором настоящей статьи [1].

Идея работы основана на применении теории так называемых монотонных систем - очень перспективного аппарата дискретной оптимизации, мало известного за пределами СССР, где он был впервые предложен и активно развивается в настоящее время [2].

Задача поиска аналогов рассмотрена применительно к анализу специальных данных /множеств объектов с булевыми компонентами/. Этот специальный тип данных имеет важное практическое значение, к нему часто сводят любые другие данные, он обеспечен методами структуризации много беднее, чем, скажем, данные числовые.

Статья построена по такому плану. Второй раздел посвящен изложению основных понятий теории монотонных систем. В третьем разделе описаны конкретные монотонные системы, на базе которых дается формальное определение понятия "ассоциативный образ" объекта.

2. МОНОТОННЫЕ СИСТЕМЫ И ИХ ЯДРА

Монотонные системы как формальные конструкции для изучения структурных свойств множества эмпирических данных были введены в [2]. В этих же работах были изучены их основные свойства.

Монотонная система определяется на произвольном конечном множестве W . На семействе всех возможных пар (i, N) , где i - элемент множества W , а N - некоторое его подмножество ($i \in N \subseteq W$), определяется вещественно-значная функция $\Pi(i, N)$, удовлетворяющая условию монотонности:

$$\pi(i, H) \leq \pi(i, H \setminus k), \quad /1/$$

для $\forall i, k \in H \subseteq W$.

На основе /1/ строится также вещественно-значная функция $F(H)$, определенная на 2^W степени множества W :

$$F(H) = \min_{i \in H} \pi(i, H). \quad /2/$$

Центральная задача, которая ставится и конструктивно решается в теории монотонных систем: поиск глобального максимума $F(H)$ на булеане 2^W :

$$F(H) \rightarrow \max_{H \in 2^W} \quad /3/$$

Любое решение задачи /3/ в соответствии с [2] называется ядром рассматриваемой монотонной системы. В [2] показано, что множество ядер монотонной системы является замкнутым относительно операции теоретико-множественного объединения семейством, и поэтому имеется только одно ядро, которое содержит все остальные ядра /наибольшее по включению ядро/. Алгоритм решения задачи /3/, предложенный в [2], гарантирует нахождение именно этого наибольшего ядра для произвольной монотонной системы за число шагов, пропорциональное $|W|^2$, причем вычислительная сложность одного шага в основном определяется объемом вычислений, необходимых для определения значения функции $\pi(i, H)$ для одной пары (i, H) . Общая идея алгоритма решения задачи /3/ хорошо поясняется следующей схемой.

Последовательно вычисляются:

$$1/ H_1 = W_1$$

$$F(H_1) = \pi(i_1, H_1) = \min_{i \in H_1} \pi(i, H_1),$$

$$2/ H_2 = H_1 \setminus i_1,$$

$$F(H_2) = \pi(i_2, H_2) = \min_{i \in H_2} \pi(i, H_2),$$

...

$$k / H_k = H_{k-1} \setminus i_{k-1},$$

$$F(H_k) = \Pi(i_k, H_k) = \min_{i \in H_k} \Pi(i, H_k)$$

...

$$n / H_n = H_{n-1} \setminus i_{n-1},$$

$$F(H_n) = \Pi(i_n, H_n) = \min_{i \in H_n} \Pi(i, H_n).$$

Здесь $n = |w|$, откуда следует, что $H_n = i_n$ и $F(H_n) = \Pi(i_n, H_n)$.

Далее в ряду чисел $\{F(H_1), \dots, F(H_n)\}$ находится наибольшее, причем если таких оказалось несколько, то выбирается число, которое было выбрано раньше всего /у которого соответствующее H_k имеет наименьший индекс "k"/. Пусть это $F(H_\mu)$. Тогда, как это доказано в [2], H_μ есть искомое, наибольшее по включению ядро данной монотонной системы /решение задачи /3//.

Если на некотором "k" шаге этой последовательности оказалось, что имеется несколько элементов i_k, i_k'', \dots таких, что любой из них дает одно и то же минимальное значение $\Pi(i_k, H_k)$, то в качестве выбираемого элемента i_k может быть взят любой из указанных элементов, например, тот, индекс которого в исходном упорядочении множества W является наименьшим.

Из указанной схемы с очевидностью следует, что особый интерес с точки зрения экономии вычислений представляют такие монотонные системы, которые допускают такое рекуррентное представление

$$\Pi(i, H \setminus k) = G(\Pi(i, H)) ,$$

когда объем вычислений для G не превышает "n" операций, каждая из которых имеет фиксированную, не зависящую от n сложность. Именно такие функции встречаются часто в приложениях [2], и именно они обеспечивают эффективность решения задачи /3/ не сложнее n^2 фиксированных вычислений.

3. АССОЦИАТИВНЫЙ ОБРАЗ ОБЪЕКТА В ИНФОРМАЦИОННОМ МАССИВЕ

Подход к выделению аналогов для данного объекта в анализируемом массиве отличается от традиционных тем, что он не базируется /как все они/ на понятии меры сходства двух объектов - данного с каким-либо из элементов анализируемого массива.

Мы исходим из того, что акт сравнения объекта с множеством с целью выделения в этом множестве элементов, похожих на данный объект, предполагает процесс оценки качества разбиения данного множества на два подмножества: подмножество похожих на данный объект, и подмножество непохожих на него.

С этих позиций следует рассмотреть пространство всех возможных разбиений данного множества объектов на два подмножества, ввести функционал над этим пространством, который играет роль искомой оценки, и построить алгоритм экстремизации этой оценки.

Реализацию такого плана проще всего осуществить, опираясь на теорию и методы кластерного анализа. Однако попытка проведения такой работы натолкнулась на два осложняющих обстоятельства.

Во-первых, известные критерии качества оцениваемых классификаций являются абсолютными в том смысле, что они задаются на том или ином семействе разбиений данного множества объектов, но никак не зависят от других, "внешних" объектов. Их использовать для указанных целей не представляется возможным.

Во-вторых, точные алгоритмы экстремизации этих критериев имеют экспоненциальную сложность, и поэтому на практике применяются, как правило, только грубо приближенные алгоритмы. В задаче поиска аналогов, особенно в тех случаях, когда анализируемое множество имеет большую мощность, такие алгоритмы могут неконтролируемо пропускать очень важные аналоги или дать в качестве таковых случайные объекты.

Наша цель состоит в том, чтобы показать, как на основе теории монотонных систем может быть построен искомый критерий.

Будем рассматривать объекты, задаваемые N -мерными булевыми векторами, компоненты которых суть свойства /или признаки/, описывающие эти объекты.

Введем в рассмотрение булеву матрицу $\phi = \|\phi_{ip}\|$, $i = \overline{1, n}$, $p = \overline{1, N}$:

$$\phi_{ip} = \begin{cases} 1, & \text{если } i\text{-й объект имеет } p\text{-е свойство,} \\ 0 & \text{- в противном случае,} \end{cases} \quad /4/$$

с помощью которой будем описывать рассматриваемое множество объектов $\{\phi_1, \dots, \phi_n\}$, где элемент $\phi_i = \{\phi_{i1}, \dots, \phi_{iN}\}$ - вектор-строка этой матрицы. Пусть $\phi_{n+1} = \{\phi_{n+1,1}, \dots, \phi_{n+1,N}\}$ - вектор строки, описывающий заданный "тестовый" объект, аналог которого необходимо выделить в матрице ϕ .

С помощью ϕ_{n+1} выделим в ϕ подматрицу $\phi^{n+1} = \|\phi_{ip}^{n+1}\|$, которая, как и ϕ , имеет n строк, но число N_{n+1} ее столбцов меньше. Она получается из ϕ удалением тех столбцов, по компонентам которых в векторе ϕ_{n+1} стоит "0". Матрица ϕ^{n+1} получается следующим образом. Сначала строится преобразование $\phi_{ip}^{n+1} = \phi_{ip} \cdot \phi_{(n+1)p}$. Затем из матрицы $\|\phi_{ip}^{n+1}\|$ с $p = \overline{1, n}$ удаляются столбцы, состоящие из одних нулей. Оставшиеся столбцы вновь перенумеровываются и теперь текущий индекс у ϕ_{ip} будет меняться в диапазоне $p = \overline{1, \dots, N_{n+1}}$.

Искомая монотонная система строится на множестве W строк полученной указанным способом матрицы ϕ^{n+1} . В качестве функции $\Pi(i, N)$ возьмем

$$\Pi(i, N) = \alpha \cdot \left| \bigvee_{j \in N} \phi_j^{n+1} \right| - \beta \left| \phi_i^{n+1} \right| \quad /5/$$

которая для других целей была введена и изучена в [3], но при дополнительных ограничениях $\alpha, \beta \geq 0$, $\alpha + \beta = 1$.

Содержательная трактовка ядра монотонной системы /5/ состоит в том, что оно объединяет объекты, максимально непохожие между собой и отличающиеся малым числом характеризующих их свойств.

Эта содержательная трактовка ядра $G(\phi^{n+1})$ индуцирует дополнительную трактовку его дополнения $\bar{G}(\phi^{n+1}) = W \setminus G(\phi^{n+1}) \cdot G(\phi^{n+1})$ - это подмножество объектов, наиболее тесно связанных между собой и отличающихся большим наличием большого числа свойств их характеризующих.

Поскольку каждый столбец из ϕ^{n+1} соответствует свойству, которое имеет место у анализируемого объекта ϕ_{n+1} , выделяемое множество $\bar{G}(\phi^{n+1})$ можно взять в качестве искомого множества аналогов данного объекта ϕ_{n+1} . Для другого объекта $\bar{\phi}_{n+1}$ даже при той же матрице ϕ будет построено другое множество $\bar{G}(\bar{\phi}^{n+1})$. Множество $\bar{G}(\phi^{n+1})$ в [1] было названо ассоциативным образом ϕ_{n+1} в множестве W , описываемым матрицей ϕ .

СПИСОК ЛИТЕРАТУРЫ

- [1] Хенчей Г.: Ассоциативно-структурный анализ булевых данных с применением монотонных систем. Автоматика и телемеханика 1987, № 5 /в печати/.
- [2] Муллат И.Э.: Экстремальные подсистемы монотонных систем I, - Автоматика и телемеханика 1976, № 5.
- [3] Кузнецов Е.Н., Мучник И.Б., Чкуаседи Н.Ф., Цварцер Л.В.: Лингвистические методы анализа распределения функций управления в организационных системах. В кн.: Анализ данных и экспертные оценки в организационных системах, М. Институт проблем управления, 1985.

РЕАЛИЗАЦИЯ СИСТЕМЫ КОНТРОЛЯ ЗА ИСПОЛНИТЕЛЬСКОЙ
ДИСЦИПЛИНОЙ НА ОСНОВЕ ПРОЦЕДУРНОЙ МОДЕЛИ

Татьяна Цекова

"Интерпрограмма", София

В настоящее время во всем мире существует тенденция увеличения числа непроизводственного персонала. В связи с этим насущной необходимостью является автоматизация административно-управленческой деятельности, которая в настоящее время обусловлена двумя противоположными тенденциями:

- увеличением расходов на учрежденческую деятельность;
- снижением цен на вычислительную технику, подходящую для целей автоматизации учрежденческой деятельности.

Основная задача административного персонала (руководства организации) - эффективное управление совокупной деятельностью соответствующей организации. Базой развития социалистической организации управленческой деятельности является высокая производственная дисциплина, точное и своевременное исполнение всех возложенных задач. В этой связи особое значение приобретает создание на базе вычислительной техники систем контроля исполнительской дисциплины (СКИД).

Традиционно СКИД реализовывались на больших

ЭВМ и задания, которые ставились на контроль, были почти никак не связаны друг с другом. Основными функциями СКВД являются [1]:

- сохранение информации о текущих задачах в организации (содержание задачи, исполнитель, возложитель, сроки и пр.);

- извещение исполнителя о возложенных ему задачах и при необходимости напоминание о приближающихся сроках;

- информация руководителей об исполнении поставленных задач и выдача различных справок об исполнительности.

Перенос функций первоначальных картотечных систем контроля исполнительской дисциплины в вычислительную среду привел к сохранению некоторых их недостатков, которые ограничивают сферу их применения. Наиболее существенные недостатки при этом следующие:

- сохраняемые данные по виду и структуре соответствуют нуждам данной организации и приспособление их к нуждам другой организации весьма сложно или даже невозможно;

- в случаях, когда нескольким различным исполнителям возлагаются задачи с одинаковым содержанием, необходимо многократное описание задачи;

- при реализации ударение часто ставится на составление отчетов, которые *post faktum* отражают исполнение или неисполнение различных задач, но это на практике никак не помогает при координации

деятельности организации;

- информационные и причинно-следственные связи между отдельными задачами не могут быть отражены в системе;

- необходимо отдельное описание каждой задачи после ее возникновения, т.е. такие системы не подходят для контроля за типовыми, повторяющимися задачами.

В результате такие системы контроля исполнительской дисциплины мало помогают при автоматизации административно-управленческой деятельности организации, причем их вклад выражается главным образом в их информационно-справочных функциях. Иногда эксплуатация таких систем требует больше ресурсов и менее надежна, чем применение простых картотечных систем, работающих без ЭВМ.

Рассматривая деятельность организации в целом, видно, что процесс ее управления состоит из отдельных действий, иногда повторяющихся, но часто взаимосвязанных, со сложной структурой связей между ними, зависящих от различных условий. Описание систем автоматизации учрежденческой деятельности (САУД) в этом случае должно отражать их специфические особенности, как функциональные (распространение данных по определенным признакам, напоминание о подлежащих выполнению работах и пр.), так и специальные типы данных (группы данных, неструктурированные и неполные данные, сложная обработка времени, роли учрежденческих служащих и др.).

В настоящее время существуют различные концептуальные модели, используемые при описании и создании САУД. Наиболее широко распространенными концептуальными моделями являются [2]:

- модели, основанные на данных;
- процедурные модели;
- функциональные модели;
- смешанные модели.

Процедурные модели наиболее естественны для реализации СКИД. Целью использования процедурной модели является представление учрежденческой деятельности скоординированным способом. При процедурных моделях анализ и описание учрежденческой деятельности основываются на идее рассмотрения различных работ, фактически выполняемых в учреждении. Представление процедур в основном базируется на использовании расширенных "условия/действия" сетей Петри.

Целью доклада является представление системы контроля исполнительской дисциплины, основанной на процедурной модели и реализованной на микро-ЭВМ - ППП "КАЛЕНДАРЬ".

ППП "КАЛЕНДАРЬ" можно рассматривать как второе поколение систем контроля исполнительской дисциплины. От своих предшественников он отличается расширенным набором понятий и функций, что в целом позволяет перейти от контроля за отдельными поручениями к комплексному контролю за функционированием организации [3].

В ППП "КАЛЕНДАРЬ" функционирование организации рассматривается как последовательность упорядоченных и взаимосвязанных действий. Основными понятиями при этом являются "работа", "действие", "процедура", "задание".

Работа - это целенаправленная, с фиксированным началом и концом деятельность, выполнение которой необходимо для функционирования организации. Работа состоит из отдельных действий, которые рассматриваются системой как элементарные, имеют ответственного (ответственных) за их выполнение и определенный срок выполнения.

Процедура - это множество упорядоченных действий, совершаемых одним или несколькими служащими для выполнения работы.

В ППП "КАЛЕНДАРЬ" действия отражаются в виде заданий.

Задание - это набор сведений, сохраняемых в системе и относящихся к конкретному действию, которое должно быть выполнено для обеспечения нормального выполнения соответствующей процедуры.

Специальным типом заданий является групповое задание - это совокупность заданий с однотипным содержанием, выданных нескольким исполнителям.

Таким образом, понятиям "работа" и "действие", отражающим функционирование организации, в ППП "КАЛЕНДАРЬ" отвечают понятия "процедура" и "задание".

Хотя в различных организациях и выполняются

одинаковые работы, как правило в каждой организации они выполняются по-своему, в зависимости от специфических условий данной организации. Преимущества использования ППП "КАЛЕНДАРЬ" проявляются именно при отслеживании специфических для данной организации стандартных процедур.

Описания отдельных процедур, так же как и некоторые сведения о состоянии организации сохраняются в информационной базе ППП "КАЛЕНДАРЬ".

Когда возникает необходимость в выполнении некоторой работы, ППП "КАЛЕНДАРЬ" активирует соответствующую процедуру - вызывает ее.

Вызов процедуры может быть автоматическим и неавтоматическим. Неавтоматический вызов выполняется пользователем по имени процедуры. Автоматический вызов выполняется ППП "КАЛЕНДАРЬ" при удовлетворении условий автоматического запуска процедуры.

В результате каждого вызова процедуры из ее описания генерируется конкретное множество заданий, называемое экземпляром процедуры.

Отдельные выполнения многократно повторяющейся работы могут отличаться некоторыми конкретными значениями отдельных данных. Значения таких данных обычно не известны при описании процедуры, они конкретизируются во время ее вызова. Такие данные называются параметрами вызова процедуры. Значения параметров вызова действительны только в рамках конкретного экземпляра процедуры. Параметры

описываются при вводе описания процедуры в ППП "КАЛЕНДАРЬ" и им могут присваиваться значения по умолчанию.

В рамках конкретного экземпляра процедуры существуют также специальные данные - локальные переменные. Локальные переменные - это числовые переменные, содержащие фактографическую информацию и получающие значения в процессе выполнения экземпляра процедуры (например, процент завершения отдельного задания).

Для описания состояния всей организации в ППП "КАЛЕНДАРЬ" существуют данные, доступные всем заданиям и процедурам. Это - глобальные переменные. Глобальные переменные могут быть отдельными значениями или массивами значений.

Параметры, локальные и глобальные переменные используются для конкретизации отдельных экземпляров процедур и для управления последовательностью выполнения заданий процедуры и отражения связей между ними.

Характерная особенность ППП "КАЛЕНДАРЬ" - использование множества дат и выражений с датами. Это является необходимым при контроле исполнительской дисциплины, для проверки наступил ли срок запуска процедуры или задания, выдачи напоминания, окончания задания. В этих случаях необходимо бывает проверить попадает ли текущая дата в определенное множество дат. Необходимые даты и их множества описываются аналитично или путем перечисле-

ния во время описания процедур.

Основными режимами работы ППП "КАЛЕНДАРЬ" являются:

- описание процедур;
- вызовы процедур;
- контроль исполнения;
- выдача справок и сводок;
- обслуживание информационных фондов.

Помимо этого существует специальный режим генерации системы, который используется только во время создания СКИД.

В режиме описания процедур выполняется ввод, корректировка и удаление их описаний. В режиме вызовов процедур осуществляется автоматический, неавтоматический и аварийный запуск процедур. В режиме контроля исполнения вводятся единичные задания и отметки об исполнении, выполняется корректировка отдельных заданий и выдача извещений и напоминаний и т. д. В режиме выдачи справок и сводок осуществляется распечатка необходимой информации о состоянии системы. В режиме обслуживания информационных фондов выполняется загрузка соответствующих фондов, их поддержка, распечатка содержащейся в них информации и т. д.

Отдельные режимы работы ППП "КАЛЕНДАРЬ" реализуются одним или несколькими функциональными модулями.

Информационная база ППП "КАЛЕНДАРЬ" состоит из следующих основных информационных фондов:

- фонда описаний процедур (ФОП);
- фонда фактографической информации (ФФИ);
- справочника структуры организации (ССО);
- фонда экземпляров процедур (фонд заданий) (ФЗ).

Эти фонды содержат следующую информацию:

- фонд описаний процедур - совокупность формализованных описаний типовых процедур. Описания используются для получения на их основе совокупностей заданий, соответствующих требуемому экземпляру процедуры;

- фонд фактографической информации - набор разнообразных данных, используемых в процессе запуска и выполнения процедур (сведения о состоянии организации, внешней среды, самой системы);

- справочник структуры организации - описание структуры организации и набор персональных данных о пользователях системы;

- фонд заданий - совокупность сформулированных заданий, которые выполняются или должны быть выполнены исполнителями. Задания в фонде заданий сгруппированы по экземплярам процедур. С каждым экземпляром связано свое множество локальных переменных и параметров вызова.

Помимо этих фондов в информационную базу данных входят: каталог базы данных, фонд бланков и очередь вызовов.

Физическая реализация информационной базы основана на использовании таблиц реляционного типа.

Структура отдельных таблиц является частично определенной, т. е. существуют обязательные данные, которые могут быть дополнены по желанию пользователей.

Упрощенная функциональная схема ППП "КАЛЕНДАРЬ" приведена на рисунке.

С точки зрения системы "КАЛЕНДАРЬ" пользователи делятся на несколько групп, играющих различные роли при взаимодействии с системой. Один и тот же пользователь в связи с отдельными заданиями может выступать в различных ролях. Роли пользователей следующие:

- распорядители - те, кто инициирует запуск той или иной процедуры, постановку на контроль некоего задания, изменение регламентов (содержания процедур) и содержания системных справочников;

- исполнители - те, кому адресованы задания и кто обеспечивает их выполнение;

- контролирующие - сотрудники, на которых возложено принятие результатов выполнения заданий. В качестве контролирующих преимущественно выступают соответствующие распорядители или прямые начальники исполнителей;

- информаторы - сотрудники, в обязанности которых входит поставка системе различных сведений о состоянии организации или внешних событиях;

- системные аналитики - специалисты, разрабатывающие регламенты, типовые процедуры и т.п.;

- администратор - сотрудник, непосредственно

работающий с системой на ЭВМ. Он отвечает за поддержку базы данных, выполняет ежедневный запуск системы или ее компонентов для запуска или контроля заданий, для выдачи необходимых справок и т.д.

Интерфейс с пользователями осуществляется посредством форматированных сообщений (формуляров, бланков), определяемых во время генерации системы.

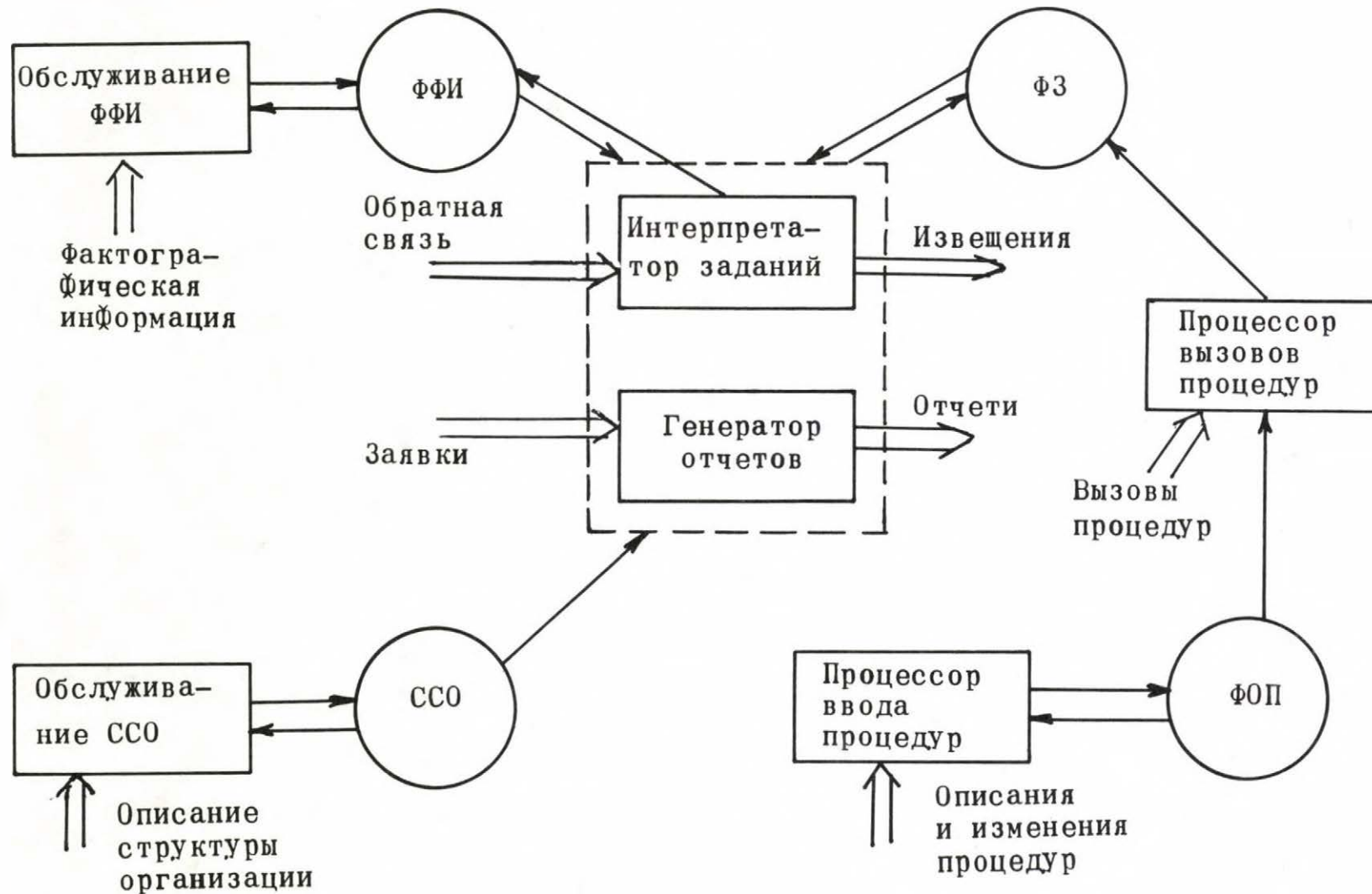
В настоящее время реализована первая версия системы. Ее пробная эксплуатация показала, что для получения большего эффекта нужно создать интегрированную систему, в которую входила бы и электронная почта. Это устранило бы существенную часть бумажных документов, используемых при функционировании СКВД.

Л И Т Е Р А Т У Р А:

1. Г. Ганчев. Програмно осигуряване на контрол на изпълнителската дисциплина. Семинар "Интерпрограмма'85". Болгария, Солнечный берег, май 1985 г.
2. Giampio Bracchi, Barbara Pernici. Methodologies and models for the design of office information systems. Joint International Symposium on Information Systems. Sydney, April, 9-11, 1984.

З.Г.Ганчев, В.Василева, А.Балаян, Т.Цекова, Г.Прохорова. Система комплексного контроля за деятельностью организации "КАЛЕНДАРЬ". "ПЕРСКОМП'85". София, апрель 1985 г.

УПРОЩЕННАЯ ФУНКЦИОНАЛЬНАЯ СХЕМА СИСТЕМЫ "КАЛЕНДАРЬ"



АБОНЕМЕНТНАЯ СИСТЕМА ДЛЯ КОРРЕСПОНДЕНЦИИ

Владимир Шкуртов, Стефан Керпеджиев

1. Введение. Появление микропроцессорных систем отмечает новый этап в развитии вычислительной техники. Широкое применение этих систем обуславливается их очевидными преимуществами: низкая цена, широкая доступность на рынке, небольшие размеры и удобство при эксплуатации. Посредством микропроцессорных систем круг применения вычислительной техники значительно расширяется. Появляется возможность ее использования в новых, нетрадиционных областях, где нельзя использовать другой вид вычислительной техники, главным образом по экономическим соображениям. Использование микропроцессорных систем, чаще всего в виде универсальных микрокомпьютеров для индивидуального потребления, позволяет осуществить повсеместную автоматизацию некоторой рутинной человеческой деятельности, для которой характерно: нерациональное использование интеллектуального человеческого труда и отсутствие гарантии верности проделанной людьми работы.

Предметом настоящего доклада является разработанная в Институте математики с ВЦ Болгарской АН система для автоматизации корреспонденции отдельных лиц и организаций. Рассмотренные возможности применения системы свидетельствуют о том, что она относится к областям нетрадиционного применения вычислительной техники. Эффективное использование системы предполагает наличие микрокомпьютера.

Несмотря на это, для более крупных организаций, располагающих миникомпьютерами, использование подобной системы расширит круг задач, решаемых с их помощью.

2. Общая характеристика системы. Рассматриваемая система автоматизирует деятельность, связанную с посылкой какой-то информации лицам или организациям по почте. Эта информация может быть произвольной и не является предметом системы. Тем не менее, при наличии автоматизированной системы синтеза документов можно осуществить ее интегрирование с рассматриваемой системой и таким образом полностью автоматизировать весь процесс подготовки посылаемой информации и ее распространения.

Почему нужна эта система и чем поможет она людям? Очень часто отдельные лица или организации вынуждены поддерживать списки адресов лиц и организаций, которым посылаются письма по определенному поводу. Часто бывает так, что одно и то же лицо подлежит уведомлению по разным поводам и поэтому фигурирует в разных списках. Это обстоятельство существенно затрудняет ручную подготовку адресов, для которой характерны следующие типичные ошибки: неправильно переписывается адрес; посылается сообщение незаинтересованному лицу; пропускается заинтересованное лицо; получаются несколько одинаковых сообщений; из-за неправильной актуализации адресов одно и то же лицо фигурирует в списках под разными адресами.

Лиц или организаций, получающих письма, будем называть абонентами, а поводы, при которых они получают письма — событиями. Отношение данного абонента к определенному событию можно выразить фла-гом булевского типа. Таким образом получается матрица абонента (рис. 1). Строки матрицы соответ-ствуют абонентам, которых обслуживает система. Каж-дый абонент представлен своим почтовым адресом. Столбцы матрицы соответствуют определенным эле-ментарным событиям.

СОБЫТИЯ

	C_1	C_2	\dots	C_n
A_1	ДА	НЕТ	\dots	ДА
A_2	ДА	ДА	\dots	НЕТ
\vdots	\vdots	\vdots		\vdots
\vdots	\vdots	\vdots		\vdots
\vdots	\vdots	\vdots		\vdots
A_m	ДА	ДА	\dots	ДА

АБОНЕНТЫ

Рис. 1

Абонементная матрица булевского типа является естественным обобщением ручного способа подго-товки адресов. Каждый столбец матрицы соответству-ет определенному списку и определяет подмножество абонентов, абонированных на соответствующее собы-тие. Булевский тип столбцов вносит и некоторые

ограничения. Чтобы расширить класс возможных применений, можно снять эти ограничения. Для этой цели используется матрица с произвольным типом столбцов: булевым, числовым и текстовым. В последних двух случаях более удачно говорить не о событиях, а о характеристиках абонента, значения которых сохраняются в абонементной матрице. В этом случае событие определяется как сравнение сохраняемого в столбце значения с константой.

В общем случае выбор абонентов, подлежащих уведомлению, осуществляется вычислением произвольной булевой функции над событиями, отраженными в абонементной матрице. Анализ многих возможных применений показывает, что чаще всего эта функция конъюнкция или дизъюнкция отдельных событий, а не их комбинации. Таким образом включение механизма вычисления произвольных булевских функций только усложняет реализацию системы, не принося реальной пользы.

3. Описание реализации. Рассматриваемая система реализована с экспериментальной целью в 1985 году для миникомпьютера CM-4 на языке Паскаль. Это благоприятствует быстрому переносу системы. Таким образом, после завершения экспериментов, стало возможным начать ее перенос на микрокомпьютеры, являющиеся функциональным эквивалентом IBM PC. Новый вариант позволит использовать преимущества микропроцессорных систем при реализации подобного рода программных систем.

Общая структура системы дана на рис. 2.

Система состоит из трех основных модулей:

- модуль для связи с потребителем;
- модуль для управления архивом;
- модуль для преобразования почтовых адресов.

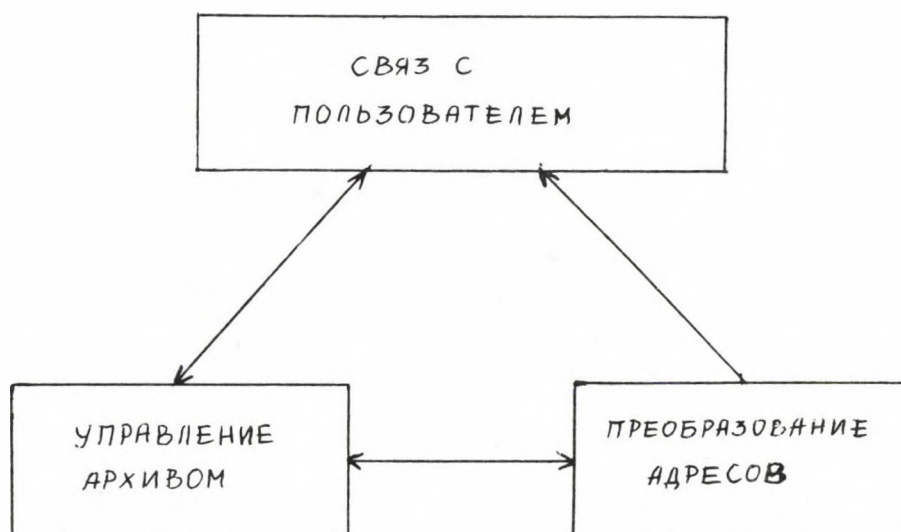


Рис. 2

Первый модуль обрабатывает заявки потребителей и осуществляет их выполнение. Второй модуль "заботится" о поддержке архива системы. Третий модуль обеспечивает поддержку различных форм почтовых адресов. С его помощью данный адрес преобразуется из одной формы в другую. В достижении главной цели системы этот модуль решающей роли не играет, но его наличие обуславливается желанием сделать общение с системой более естественным для пользователя. Специфические особенности различных форм почтовых адресов и их преобразования даны ниже.

Структура архива системы такая, что допускает реализацию сравнительного большого числа разных микропроцессорных систем. В большинстве случаев микропроцессорные системы снабжены запоминающими устройствами небольшой емкости, поэтому приняты для сравнительно экономной реализации. Архив системы состоит из двух частей: матрица абонента и список почтовых адресов абонентов. Почтовые адреса сохраняются вне матрицы, потому что они, как самостоятельные объекты представляют собой сравнительно сложные, в общем случае нестандартизованные конструкции. Матрица абонента реализована как последовательный файл с возможностью прямого доступа к отдельным записям. Число и тип столбцов фиксированы для того, чтобы было возможно изобразить одну строку матрицы в одной записи фиксированной длины. Это ведет к необходимости дополнительного настраивания системы при каждом конкретном применении.

Основные операции системы можем разделить условно на две группы:

- операции, связанные с автоматизированием выбора абонентов, подлежащих уведомлению;
- операции, связанные с автоматизированием процесса ввода, вывода и поддержки почтовых адресов и дополнительной информации, связанной с ними в абонентной матрице.

Для выбора абонентов, подлежащих уведомлению, необходимо указать наступление определенного события. В соответствии с данными выше примечаниями,

каждое событие является конъюнкцией или дизъюнкцией определенных потребителем элементарных событий. Для столбцов небулевского типа элементарные события в абонементной матрице определяются константой и видом совершаемого сравнения. Осмотрев абонементную матрицу, система определяет абонентов, подлежащих уведомлению и предоставляет потребителю необходимую адресную информацию, оформленную в подходящем виде.

Так, как абонементная матрица состоит из числовых, булевских и строковых значений, операции, связанные с ее поддержкой, сравнительно простые. Эти операции следующие: включение абонента, удаление абонента и изменение информации, связанной с данным абонентом. Они реализуются соответствующими операциями над строками абонементной матрицы. Абоненты определяются по их идентификационному номеру, присвоенному системой при включении абонента. Благодаря наличию модуля для преобразования почтовых адресов, становится возможным определять абоненты по их адресам, которые могут быть и неполными (например, только по имени абонента).

Для увеличения гибкости системы в будущем предусматриваются операции по включению нового столбца в абонементную матрицу и удалению столбца из нее. Первая из них сравнительно сложная, из-за того, что требует немедленного ввода информации для всех существующих в системе абонентов.

Операции, связанные с поддержкой адресов абонентов, значительно сложнее. Затруднения свя-

заны с отсутствием стандартизации по отношению почтовых адресов.

4. Специфические особенности представления адресов и их преобразований. Для обеспечения эффективной работы системы, в ней поддерживаются три представления почтовых адресов:

— внутреннее представление — Оно служит для сохранения адресов в архиве и поэтому должно быть экономным. С целью осуществления контроля структуры адресов и их правильного сравнения, это представление спроектировано так, что содержит почтовый адрес в анализированном виде:

— свободное представление — Оно используется при вводе адресов потребителями системы и включает наиболее часто используемые людьми средства для записи почтовых адресов. Для осуществления автоматического преобразования адресов с свободного формата во внутренний, система использует отдельно созданное формализованное описание понятия "почтовый адрес":

— внешнее представление — Оно служит для вывода адресов в различных форматах в соответствии с предназначением выхода: для почтового конверта; для включения в список, в котором каждый адрес содержит только имя и номер телефона; для быстрой справки (адрес выводится на экран в структурированном виде) и др.

На рис. 3 показаны основные виды операций над адресами, осуществляемые системой.

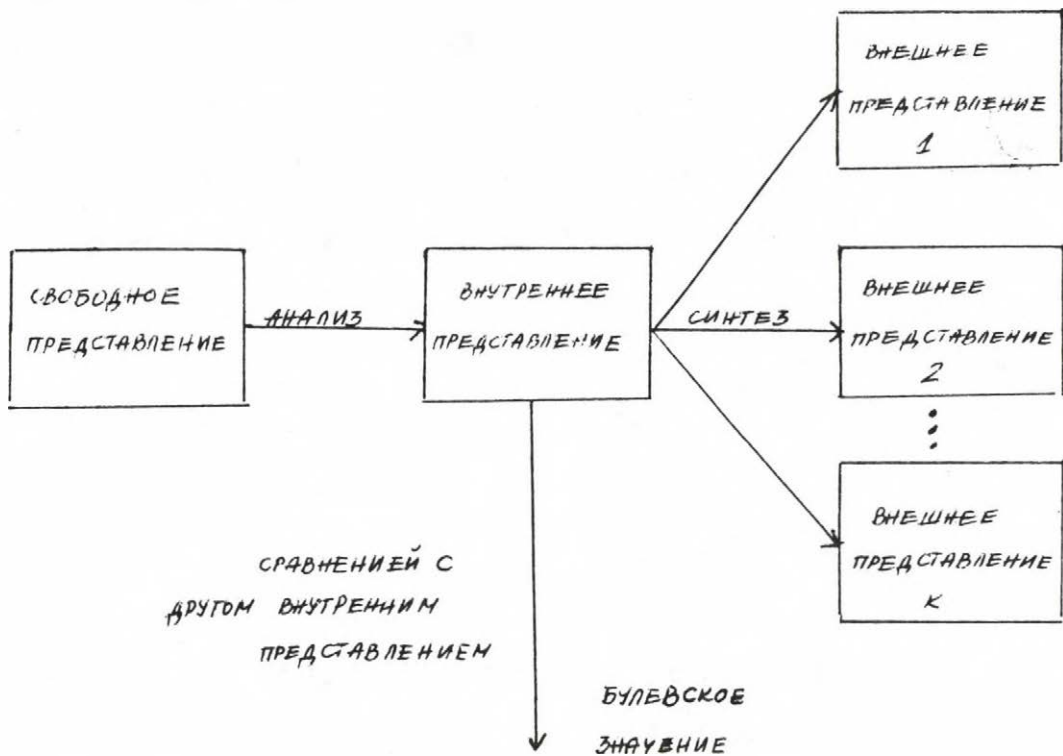


Рис. 3

а) Анализ. Эта операция производится согласно описанному в [1, 2, 3] методу, который основывается на формальном лексическом, синтаксическом и структурном описании свободных представлений адресов. Существует язык, на котором формальное описание создается независимо от рассматриваемой системы. Алгоритм анализа универсален и управляется созданным формальным описанием. Этот подход позволяет изменять свободный формат адресов, не затрагивая систему. Существующее описание позволяет системе воспринимать адреса на болгарском языке, имеющие структуру болгарских почтовых адресов. Конечно, в других условиях (например, в других стра-

нах) надо создать другое описание свободного формата адресов.

б) Сравнение. Аргументы этой операции — адрес и адрес-образец, представленные внутренним форматом. Сравнение успешно, если адрес имеет все элементы адреса-образца. Например, сравнение адреса:

София — 1156, бул. Руски 23

Иван Петров Иванов, тел. 15-60-28

с адресом-образцом:

София, Иван Иванов

успешно, пока сравнение этого же адреса с адресом-образцом:

София, бул. Руски 23, Иван Петров Николов
неуспешно, так как фамилии адресантов не совпадают.

в) Синтез. Синтез внешних представлений тоже производится универсальным алгоритмом, настроенным к конкретным внешним форматам. Это позволяет сравнительно легко расширять арсенал внешних представлений. Так например, внешние представления рассматриваемого адреса в трех упомянутых выше внешних форматах выглядят так:

— для почтового конверта:

п.к.1156, гр. София

бул. Руски 23

Иван Петров Иванов

— как элемент списка:

Иван Петров Иванов — 15-60-28

— для быстрой справки:

п.к. 1156	бул.Руски 23	Иван Петров	15-60-28
гр. София		Иванов	

5. Возможные применения системы. Как было упомянуто в начале, разработанная абонементная система для корреспонденции имеет широкий круг применения. По этой причине здесь даны только три типичные случая применения: книжные магазины для многотомных изданий, организация научных мероприятий и использование в системе здравоохранения.

Отпечатывание многотомного издания обычно продолжается долгое время. Чтобы данный покупатель мог приобрести все тома издания, книжные магазины, продающие многотомные издания, предоставляют своим клиентам абонементную форму обслуживания. Это - типичное применение разработанной абонементной системы для корреспонденции. Здесь абоненты - это клиенты книжного магазина, а событие - получение очередного тома издания, о котором клиенты должны быть уведомлены. Обычно один клиент абонируется на несколько изданий. Также часто получают одновременно томы нескольких изданий. Полученная матрица абонента типично булевского типа. Так, как система поддерживает фиксированное число столбцов, при открытии новой подписки приходится использовать свободный к этому моменту столбец. Закончив отпечатывание определенного издания, столбец, соответствующий ему, объявляется свободным. В этом случае могут появиться клиенты неабонированные ни на какое другое издание. Таких клиентов можно исключить из

системы, но все-таки лучше оставить их для уведомления об открытии новых подписок. В настоящий момент ведутся переговоры для использования разработанной абонементной системы в некоторых книжных магазинах для многотомных изданий в г. Пловдиве.

Организация научных мероприятий предусматривает уведомление потенциальных участников. Они определяются на основе их научных интересов, возраста, профессии, места работы и др. На первый взгляд в данном случае необходимо отразить в абонементной матрице характеристики абонентов (потенциальных участников) вместо определенных событий. Имея ввиду, что номенклатура отдельных характеристик очень небольшая, иногда можно снова использовать матрицу булевского типа. Например, на месте одного столбца с характеристикой "ученая степень", можем иметь отдельные столбцы булевского типа - "кандидат наук" и "доктор наук". В этом случае число столбцов матрицы увеличивается, но они становятся однотипными и с ними легче работать. Рассматриваемая система использована при уведомлении участников в национальном семинаре по информатике, организованном Союзом математиков в Болгарии.

Последняя рассматриваемая область применения предусматривает использование системы для наблюдения состояния диспансеризованных пациентов в районной поликлинике. В этом случае абоненты - сами пациенты, и приходится сохранять их характеристики - болезнь, возраст, дата последнего осмотра, состояние пациента, результаты определенных

исследований и т.д.

Чтобы удовлетворить необходимым требованиям, абонементная матрица не может быть полностью булевского типа. Использование разработанной системы задумано как часть более обширной системы для автоматизации административно-управленческой деятельности в районных поликлиниках.

6. Заключение.

В этой работе описана система, которая предназначена для автоматизации типичной конторской деятельности, а именно: организация корреспонденции с большим числом адресантов. Применение системы основано на использовании микропроцессорной техники.

Первый вариант системы, выполненный в 1982 году, как и руководство по реализации двух из модулей ("связь с пользователем" и "поддержка архива") описываемой системы, осуществил В. Шкуртов. Модуль преобразования почтовых адресов разработал и реализовал С. Керпеджиев. Часть программных реализаций сделали И. Бозов и Н. Попов. Вся работа была выполнена по инициативе и под руководством проф. П. Барнева. Всем этим людям авторы выражают свою глубокую благодарность.

ЛИТЕРАТУРА

1. П. Барнев, С. Керпеджиев, Подход к вводу данных в свободном формате примененный для дат и почтовых адресов. Математика и математическое образование, БАН, 1984, с. 244-255

2. P. Burnev, S. Kerpedjiev, Automatic Data Extraction from Specialized Texts, (предоставленная для публикации)
3. S. Kerpedjiev, A Technique for Data Extraction from Specialized Texts, (предоставленная для публикации)

1986-BAN EDDIG MEGJELENTEK:

- 179/1986 Terlaky Tamás: Egy véges criss-cross módszer és alkalmazásai
- 180/1986 K.N. Čimev: Separable sets of arguments of functions
- 181/1986 Renner Gábor: Kör approximációja a számítógépes geometriai tervezésben
- 182/1986 Proceedings of the Joint Bulgarian-Hungarian Workshop on "Mathematical Cybernetics and Data Processing" Scientific Station of Sofia University, Giulecica /Bulgaria/, May 6-10, 1985 /Editors: J. Denev, B. Uhrin/ Vol I
- 183/1986 Proceedings of the Joint Bulgarian-Hungarian Workshop on "Mathematical Cybernetics and Data Processing" Scientific Station of Sofia University, Giulecica /Bulgaria/, May 6-10, 1985 /Editors: J. Denev, B. Uhrin/ Vol II
- 184/1986 HO THUAN: Contribution to the theory of relational databases
- 185/1986 Proceedings of the 4th International Meeting of Young Computer Scientists IMICS'86 /Smolenice, 1986/ /Editors: J. Demetrovics, J. Kelemen/
- 186/1986 PUBLIKÁCIÓK - PUBLICATIONS 1985
Szerkesztette: Petróczy Judit
- 187/1986 Proceedings of the Winter School on Conceptual modelling /Visegrád, 27-30 January, 1986/ /Editors: E. Knuth, A. Márkus/

- 188/1986 Lengyel Tamás: A Cluster analízis néhány kombinatorikai és valószínűség-számítási problémája
- 189/1986 Bernus Péter: Gyártórendszerek funkcionális analízise és szintézise
- 190/1986 Hernádi Ágnes: A típus fogalma, és szerepe a modellezésben
- 191/1986 VU DUC THI: Funkcionális függőséggel kapcsolatos néhány kombinatorikai jellegű vizsgálat a relációs adatmodellben
- 192/1986 Márkusz Zsuzsanna: P a p e r s o n M a n y - s t o r e d
logic as a tool for modelling

